FEBRUARY 1990 £2.75

Commodore DISK USER

THE 1st MILLION

The Strategist
Tomorrow's Tomorrow
Problems solved

ISSN 0953-0614

DATEL ELECTRONICS

DATA RECORDER

- Quality Commodore compatible data recorder.
- Pause control, counter, etc.
- Suitable for 64/128.
- Send now for quick delivery.

ONLY £24.99

SPRINT 128

- Full feature Centronics Printer Interface.
- Connect your 64/128 to a range of full size Centronics Parallel Printers.
- Easy to use - supports Commodore Graphics Set.
- Onboard Microprocessor Chip means no power programmes to load.
- Works with most applications.
- No more to buy!

ONLY £29.99

PARALLEL PRINTER CABLE

- Connects full size printers to the parallel port of your C64/128.
- Many programmes and cartridges (Action Replay/Final Cartridge etc.) will drive printers from this port.

ONLY £12.99 COMPLETE

ROBOTEK 64

SPECIAL VALUE

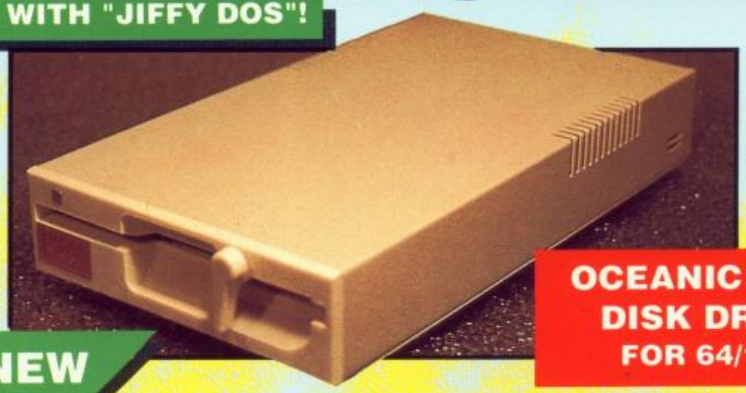
MODEL & ROBOT CONTROL MADE EASY.

- 4 output channels - each with onboard relay.
- 4 input channels - each fully buffered TTL level sensing.
- Analogue input with full 8 bit conversion.
- Voice input for voice control.
- Software features:- test mode/ analogue measurement/voice activate/digital readout etc.

ONLY £39.99

INCLUDING HARDWARE/SOFTWARE/ MIC. ETC,(STATE TAPE OR DISK).

NOW WITH "JIFFY DOS"!



OCEANIC 118N DISK DRIVE FOR 64/128

NEW

The Oceanic 118N is a superb quality Disk Drive specially designed for the Commodore 64/128.

Just look at these features...

- Slimline design - very compact.
- External Power Pack so no overheating problems (unlike some types).
- Ready to go - no more to buy.
- Direct drive motor for super quiet operation and extra long life.
- Now probably the best selling replacement Disk Drive for the 64/128.
- Comes complete with manuals, connecting leads etc.

ONLY £129.99 INCLUDING FREE OCP ART STUDIO!!

FREE COPY OF OCP ADVANCED ART STUDIO (WORTH £29.99) WHILE STOCKS LAST

DIGITAL SOUND SAMPLER

- The new sampler allows you to record any sound digitally into memory & then replay it with astounding effects.
- Playback forwards/backwards with echo/reverb/ring modulation.
- Now with full sound editing module to produce outstanding effects.
- Full 8 bit D to A & ADC conversion.
- MIDI compatible with suitable interface. (i.e. Datel unit for £29.99, see ad.).
- Live effects menu includes real time display of waveforms.
- Line in/mic in/line out/feedback controls.
- Powerfull sequencer with editing features.
- Load/save sample. Up to 8 samples in memory at one time.
- Complete software/hardware package. Tape or Disk (please state).

ONLY £49.99

COM-DRUM

DIGITAL DRUM SYSTEM

- Now you can turn your digital sound sampler into a digital drum system.
- 8 digital drum sounds in memory at one time.
- Complete with 3 drum kits. Real drum sounds - not synthesised.
- Create superb drum rhythms with real & step time.
- Full editing. Menu driven. Load/ Save facilities.
- Output to hi-fi or through TV speaker.

ONLY £9.99

STATE TAPE OR DISK

RESET CARTRIDGE

- Unstoppable reset button. NOTE:- user port cheaper type reset buttons offered by others are not unstoppable.
- Resets even so called "unstoppable" programs.
- Add pokes from magazines etc.
- Simply plugs in to cartridge port.

ONLY £5.99

3 SLOT MOTHERBOARD

SAVE WEAR & TEAR ON YOUR EXPANSION PORT

- Will accept three cartridges on it's high grade PCB.
- Switch in/out any slot.
- Fully buffered.
- Reset button and an onboard safety fuse.

ONLY £16.99

TOOLKIT IV

THE ULTIMATE DISK TOOLKIT FOR THE 1540/1541.

- A disk toolkit is an absolute must for the serious disk user. Toolkit IV has more features than most for less.
- DISC DOCTOR V2 - Read & write any track & sector including extra & renumbered tracks. Repair damaged sectors.
- HEADER/GAP EDITOR - Decodes & displays ALL header information including off bytes & header gap. Rewrite the entire header & header gap. Renumber sectors. Also edit any sector tail gap.
- DISK LOOK - Sort directory. Recover lost files. Display file start /end addresses. Disassemble any file program directly from the disk to SCREEN or PRINTER including undocumented opcodes. Edit Bam.
- FAST FILE COPY - Selective file copy. Works at up to 6 times normal speed.
- FAST DISK COPY - Copy an entire disk in 2 minutes or less using single 1541.
- FILE COMPACTOR - Can compact machine programs by up to 50%. Save disk space.
- FORMATTER - 10 second format an entire disk or format any individual track or half track 0 to 41.
- ERROR EDIT - Quickly find & recreate all read errors including extras & renumbered tracks or sectors & half tracks from 0 to 41.

ONLY £9.99

DATEL ELECTRONICS

UNBEATABLE MIDI PACKAGE!!

SAVE £30

DATEL MIDI 64 INTERFACE PLUS ADVANCED MUSIC SYSTEM ONLY £39.99

NO MORE TO BUY - THE TOTAL MIDI CONNECTION

FREE MIDI CABLES

The Advanced Music System is probably the best MIDI/music package ever produced for the 64/128 offering a huge range of musical composition features plus MIDI compatibility - add the Datel MIDI Interface and you have the TOTAL MIDI SOLUTION!!

- **EDITOR MODULE...** just like a word processor for music. All the features you would expect.
- **MIDI MODULE...** this is the module which allows the full potential of the Music System and your MIDI keyboard to be achieved. Using the Datel MIDI 64 Interface any MIDI instrument (including Yamaha - see below) can be connected to your 64.
- **SYNTHESISER MODULE...** probably the most powerfull module. Create sounds with full waveform editing, realtime sequencing etc.
- **PRINTER MODULE...** allows you to print out your music to a range of printers including Commodore and Epson compatibles. Printout can be edited and can also include lyrics if required!
- **LINKER MODULE...** allows large musical compositions to be created from up to 26 files linked together - offering Tempo and Time Signature adjustments.
- **HUGE RANGE OF FEATURES...** Advanced Music System has literally hundreds of commands and features - we have only outlined the main headings - this is a truly professional package.
- **KEYBOARD MODULE...** allows for entry of music from the QWERTY keyboard. Sequencer works like digital recorder.

BUT THAT'S NOT ALL...

...WE CAN OFFER THE ABOVE MIDI PACKAGE TOGETHER WITH THE SUPERB YAMAHA SH10 MIDI SYNTHESISER FOR AN UNBEATABLE PRICE!!

UNBEATABLE OFFER

Just look at these features...

- The world's first shoulder keyboard.
- With 25 built-in instrument choices.
- Choice of 25 rhythm styles.
- Vibrato, sustain and portamento special effects available.
- 3 fill-in variations for professional rhythm changeovers.
- Complete range of 32 keys for great melody play-a-long.
- Compose your own backing into the chord sequencer.

ONLY £99.99

SYNTHESISER, MIDI 64 INTERFACE AND ADVANCED MUSIC SYSTEM PLUS FREE MIDI CABLES - NO MORE TO BUY!!

VIC 20 RAMPACK

- Simple plug-in memory expansion cartridges for your VIC 20.
- Never be short of memory again!
- Two models available - 16K or 32K.
- Both models are switchable for various memory settings i.e. 3K, 16K, 24K, etc.
- Many VIC programs require extra RAM - this is your answer.

ONLY £14.99 FOR 16K

ONLY £19.99 FOR 32K

NOW YOU CAN INCREASE THE MEMORY OF YOUR C16 EASILY

C16 RAMPACK

- Plug in RAM cartridge gives you up to 16K EXTRA MEMORY!!
- That's over 28K free for Basic and even an extra 18K in HiRes mode!
- Plugs in to memory expansion port - then just switch on.

ONLY £14.99

EPROM ERASER

- This handy AC/DC Eprom Eraser will erase up to 4 chips quickly and easily.
- Built-in timer and safety switch on cover.
- Superbly styled - small enough to fit in pocket - even as it works!!
- Works from built-in battery (supplied) or from AC/DC adaptor if required.

ONLY £39.99

LIMITED NUMBERS AT THIS AMAZINGLY LOW PRICE!!

256K SUPEROM EXPANDER

- Select instantly from 8 sockets which accept up to 32K EPROM each.
- Program your own EPROMs using our EPROM programer.
- No need to have lots of cartridges - just make a selection from the Superom menu.
- Fully menu driven on power up.
- Select any slot under software controls.
- Unique EPROM generator feature will take your own programs - basic or m/c & turn them into autostart EPROMs. (EPROM burner required).
- Accepts 2764/27128/27256 EPROMs.
- On board unstoppable reset.
- On board operating systems - no programs to load.

ONLY £29.99

EPROMMER 64

- A top quality, easy-to-use EPROM programer for the 64/128.
- Fully menu driven software/ hardware package makes programing/reading/verifying/ copying EPROMs simplicity itself.
- Will program 2716, 2764, 27128 & 27256 chips, 12.5, 21 or 25 volts.
- Fits into user port for maximum compatibility with cartridges/ Superom Board etc.
- Full feature system - all functions covered like device check/verify.
- We believe Eprommer 64 is the most comprehensive, most friendly & best value for money programer available for the 64/128.
- Ideal companion for Superom Board, Cartridge Development System, our kernal expanders or indeed any EPROM base project.
- Comes complete with instructions - plus the cartridge handbook.

ONLY £39.99 COMPLETE

ALL ORDERS NORMALLY DESPATCHED WITHIN 48 HRS

HOW TO ORDER ...

BY PHONE	BY POST	FAX
0782 744707 24hr Credit Card Line	Send cheques/POs made payable to "Datel Electronics"	0782 744292 UK ORDERS POST FREE EUROPE ADD £1 OVERSEAS ADD £3

PRICES AND SPECIFICATIONS CORRECT AT TIME OF PRESS AND SUBJECT TO CHANGE WITHOUT NOTICE

CALLERS WELCOME - Please reserve goods by telephone prior to visit.

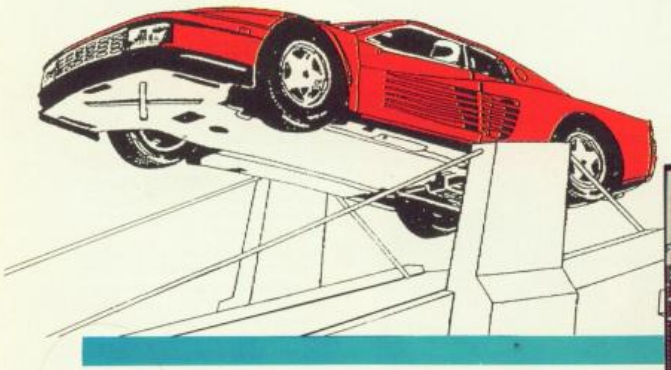
DATEL ELECTRONICS

DATEL ELECTRONICS LTD., FENTON INDUSTRIAL ESTATE
GOVAN ROAD, FENTON, STOKE-ON-TRENT, ENGLAND.

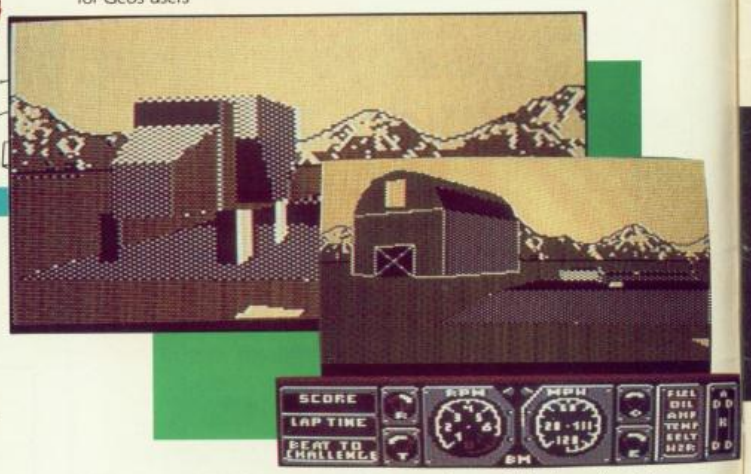
SALES ONLY	TECHNICAL ONLY
0782 744707	0782 744324

CONTENTS

IN THE MAGAZINE



ON THE DISK



SUBSCRIPTION RATES

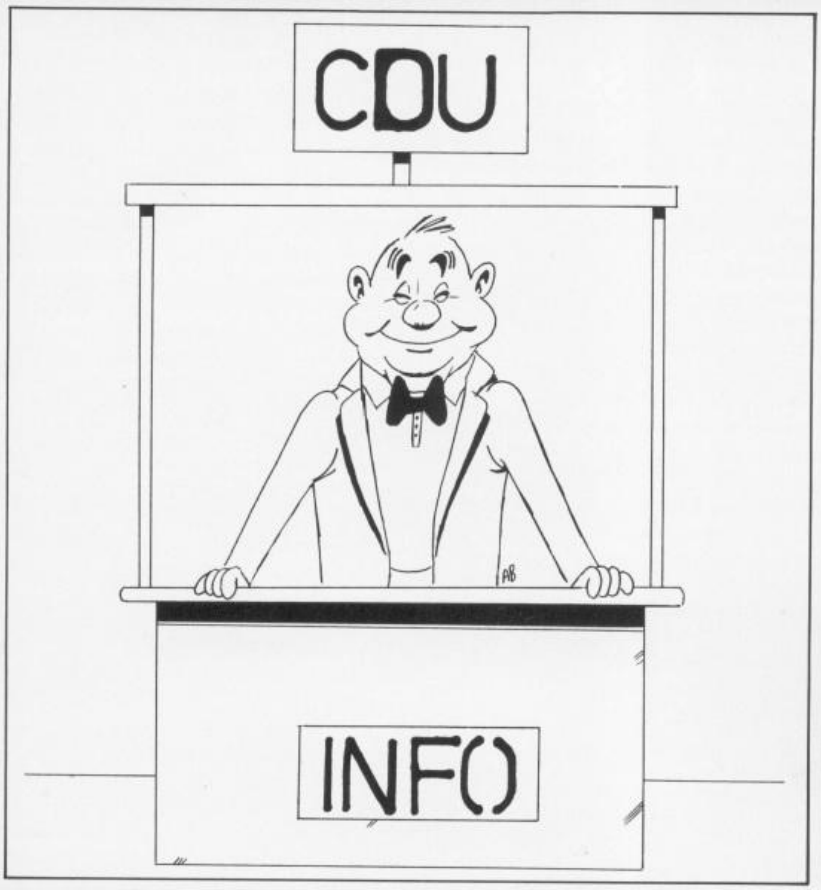
Here are the rates for subscriptions to CDU with effect from November 1989

UK	**£33.00**
Europe	**£39.00**
Middle East	**£39.30**
Far East	**£41.60**
Rest of the World £39.70 or	
USA	**$69.00**

Airmail rates on request

Editor: PAUL EVES
Group Editor: STUART COOKE
Production Editor: HILARY CURTIS
Cartoonist: ALAN BATCHELOR
Photography: MANNY CEFAI
Adventure Correspondent: GORDON HAMLETT
Advertisement Manager: PAUL KAVANAGH
Display Sales Exec: MARIA WADE
Classified Sales Exec: TONY FLANAGAN
Designer: MARK NEWTON
Origination: EBONY
Distribution: S.M. DISTRIBUTION
Printed By: Chase Web, Plymouth

Commodore Disk User is a monthly magazine published on the 3rd Friday of every month. **Argus Specialist Publications, Argus House, Boundary Way, Hemel Hempstead, HP2 7ST. Telephone: (0442) 66551 Fax: (0442) 66998.**



Editor's Comment

Here we are in the middle of November, yet before you read this, Christmas will have been and gone, as will the New Year. It's a strange feeling, trying to be topical two months before the event. Asking people how their festivities went, long before they have them is akin to reading the future. Therefore in this issue, you will see an article that makes full use of this phenomenon. (Look out for Tomorrow's Tomorrow). I therefore sincerely hope that everyones Christmas and New Year lived up to your expectations, whatever they were. A few of you appear to be experiencing some difficulties with the CDU disks. The main area of complaint is that they will just not load at all. Unfortunately this is caused by outside sources that we at Argus have no control over. Every single one of the CDU disks is analysed before it leaves the disk duplicators. Therefore we can guarantee that no faulty disks get put onto the magazines. The problems seem to arise either A) at the newsagents or B) in the post. Whilst the magazines are held in storage at the newsagents, they tend to get stacked up too compactly. Consequently the disks become tight in their sleeves thus preventing them from spinning freely. A little surgery with a razor blade will alleviate this. Once the disk becomes readable you can then copy the files onto a clean disk. The second problem is not recoverable though. The post office appear to have a habit of scanning packages for security reasons, and as a result of these scans the disks become corrupt. There is, unfortunately, no remedy to this. Argus apologise to all it's readers that suffers from one of the above. Hopefully some sort of cure will be discovered in time.

One point I must make clear is that the use of 'Fast Loaders', 'Cartridges' or alternative operating systems (Dolphin DOS) may not guarantee that your disk will function properly. If you use one or more of the above and you have difficulties, then I suggest you disable them and use the computer under normal, standard conditions.

Disk Instructions

We do our best to make sure that Commodore Disk User will be compatible with all versions of the C64 and C128 computers.

Getting the programs up and running should not present you with any difficulties, simply put your disk in the drive and enter the command.

LOAD"MENU",8,1,

Once the disk menu has loaded you will be able to start any of the programs simply by pressing the letter that is to the left of the desired program.

It is possible for some programs to alter the computer's memory so that you will not be able to LOAD programs from the menu correctly until you reset the machine. We therefore suggest that you turn your computer off and then on before loading each program.

How to copy CDU files

You are welcome to make as many of your own copies of Commodore Disk User programs as you want, as long as you do not pass them on to other people, or worse, sell them for profit.

For people who want to make legitimate copies, we have provided a simple machine code file copier. To use it, simply select the item FILE COPIER from the main menu. Alternatively, you can load it by typing LOAD'FILE COPIER',8,1 then SYS52000. The copier works with a single drive, and is

controlled by means of the function keys as follows:

F1: Copy file-the program will promt you for a filename.
F3: Resave the memory buffer-you may get an error on a save (perhaps you left the drive door open). Use this to try again. Or if you want to make multiple copies to other disks.
F5: Disk commands allows you to enter any regular C64 disk command.
F7: displays the disk directory
F2: Exits the program and returns you to basic.

Disk Failure

If for any reason the disk with your copy of CDU will not work on your system then please carefully re-read the operating instructions in the magazine. If you still experience problems then:

1) If you are a subscriber, return it to:
INFONET LTD
5 River Park Estate
Berkhamsted
HERTS. HP4 1HL
Tele: 0442-876661

2) If you bought it from a newsagents, then return it to:
CDU Replacements
Protoscan
Burrell Road
St. Ives
Cambs
PL17 4LE
Tele:0480-495520
(Within eight weeks of publication date disks are replaced free).

After eight weeks a replacement disk can be supplied from Protoscan for a service charge of £1.00. Return the faulty disk with a cheque or postal order made out to Protoscan and clearly state the issue of CDU that you require. No documentation will be provided.

Please use appropriate packaging, cardboard stiffener at least, when returning disk. Do not send back your magazine-only the disk please.

NOTE: Do not send your disks back to the above if its a program that does not appear to work. Only if the DISK is faulty. Program faults should be sent to the editorial office marked FAO bug-finders. Thank you.

Back Issues

Back issues of COMMODORE DISK USER are available at £3.25 per issue, which includes postage and packing via:
Infonet Ltd
5, River Park Estate
Berkhamsted
Herts
HP4 1HL
Tele: 0442-876661

VOL 2 No.4 MAY/JUN 89
BASE ED - Get organised with the C64 database.
DBASE 128 - 40 or 80 column storage for C128 owners.h
6510+ – The ultimate in C64 assembly programs.
SID SEQUENCER – Make Commodore music with ease.
LIBERTE – Escape the POW camp in this 1940's style adventure.
FX KIT – Bangs, Pows and Zaps made easy.

VOL 2 NO.5 JUL/AUG 89
FONT FACTORY – Create your own chars.
HI-RES DEMO KIT – Add music to your favourite picture.
ANIMATOR – Get those sprites moving.
BORDER MESSAGE SCROLL – Say what you want along the bottom of the screen.
TYPIT 128 – Create professional text layouts on your C128.
SCREEN COPIES UTILITY – Download your favourite screens.
VIDI-BASIC – Graphic based extension to Basic.
64 NEWS DESKS – Become a C64 reporter.

VOL 2 No.6 SEP/OCT 89
MICKMON – An extensive M/C monitor.
SCRAPBOOK – Collectors and hobby-ists database.
CELLRATOR – Enter the caves if you date.
RAINBOW CHASER – Rainbows means points in the unusual game.
HIDDEN GRAPHICS – Utilise those graphic secrets.
FORTRESS – Save the world. Yet again!!
DISK HUNTER – Keep tabs on your disk library.
SUPERFILE – One more for the record keepers.

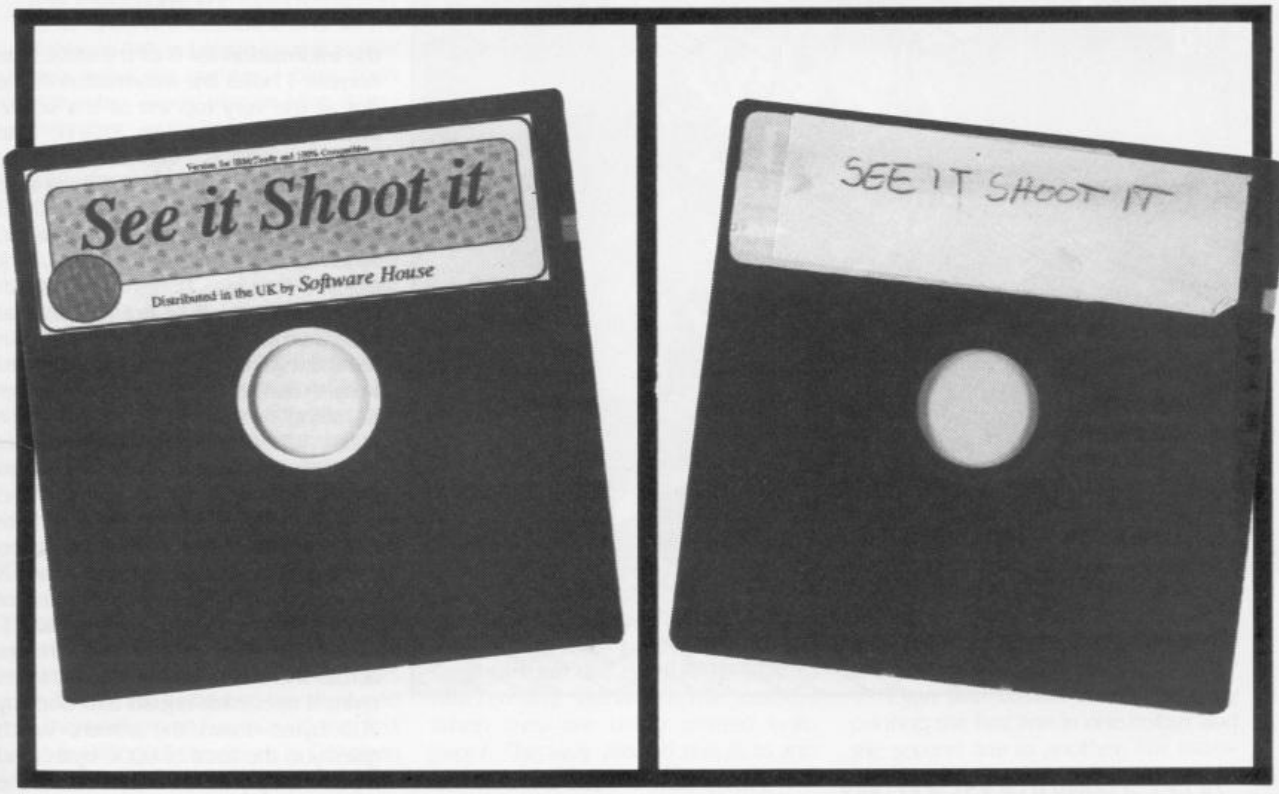
VOL 3 No.1 NOVEMBER 89
BASIC EXTENSION – Windows and Icons the easy way
B-Raid – Vertical scrolling shoot 'em up
DISKONOMISER – Prudent disk block saving
HELP – Design your own information help screens
ORSITAL – An arcade style game with a difference
PROGRAM COMPARE – Modifying Basic progs has never been easier
RASTER ROUTINES – A few colourful demos
SPRITE EDITOR 1 – A no nonesense basic sprite editor
WABBIUT – Help the rabbit collect his carrots

VOL 3 No.2 DECEMBER 89
CDU MENU KIT – Make your own colourful menus
MUSIBASIC – Sound and Music made easy
TEMPLATE DESIGN – Design your very own backdrops
QUICKWORD – A Basic wordprocessor that is expandable
KRON – Rescue the princess in this fantasy adventure
LIMBO – Collect the cells off the blocks
PHOBOS – Break out of jail and gain your freedom
PANIC – A quick eye as well as a quick joystick is required here

VOL 3 No.3 JANUARY 90
4 IN A ROW – Connect a row of counters
FROGS IN SPACE – Leap to safety across the space lanes
BLACKJACK – Don't lose your shirt
LORD OF DARKNESS – Defeat the evil lord true adventure style
MARGO – Fly around and collect the jewels
JETRACE 2000 – Have you got what it takes to be best
ULTIMATE FONT EDITOR – Create your own screens and layouts
SELECTIVE COLOUR RESTORE – Design your own start up colours
6510+UNASSEMBER – Transform M/C into Source, with labels
TRIVIA CHALLENGE – The first of 3 files for this superb game
GEOS FONTS – The first 4 of 12 fonts for Geos users

All orders should be sent to INFONET LTD, 5, RIVER PARK ESTATE, BILLETT LANE, BERKHAMSTED, HERTS. HP4 1HL
Please allow 28 days for delivery.

WARNING



THIS DISK WILL GIVE YOU HOURS OF FUN

THIS DISK COULD GIVE YOU 6 MONTHS IN PRISON

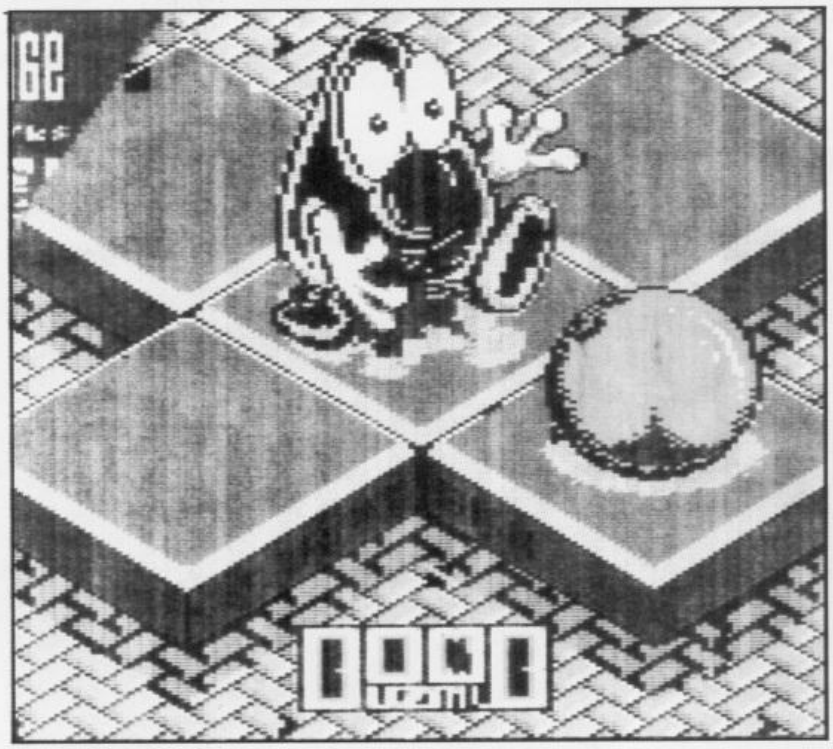
If you Pirate Software you are a thief. Thieves will be prosecuted.

THIS CAMPAIGN IS ORGANISED BY
ELSPA
EUROPEAN LEISURE SOFTWARE PUBLISHERS ASSOCIATION

Any information on piracy should be passed to The Federation Against Software Theft. Telephone 01-240 6756

PIRACY IS THEFT

Colour Picture Printing



At last, a utility that lets you print out your masterpieces created by CDU Paint, and other similar packages.

By Keith Mcleman

Before we go into detail about the program you are about to use, a little background knowledge on Bit-mapped graphics is required.

When you first switch on your computer, you can type letters or numbers or any of the other keyboard characters straight on to the screen. The computer allows 40 such characters across the screen and 25 characters down the screen, therefore allowing 1000 characters to appear altogether. Hidden in the depths of the computers memory is a 1000 byte data block which stores the position of the characters that you type in so that the computer always prints the characters in the right place on the screen. This 1000 byte data block is known as the SCREEN MEMORY.

Now, each of these characters can be printed in any one of 16 different colours, so the computer needs another 1000 byte data block to remind it of the colour that each character should be displayed in. This 1000 byte data block is known as the COLOUR MEMORY.

Each of the characters which can appear on the screen (the letter 'A' for example) is made up on an 8 dot wide by 8 dot deep grid, a total of 64 dots. As each of the 1000 characters on the screen are made up of 64 dots, you can now realise that what you see on your T.V. is actually a picture created using 64000 tiny dots (1000 characters times 64 dots in each character). Hi-res mode lets you control every one of these 64000 dots, thereby allowing you total freedom to create as complex a picture as you wish.

For hi-res pictures, the computer sets aside 8000 bytes of memory, which tells it what to display on the screen. There are 64000 dots on the screen but only 8000 bytes of data in memory, therefore each byte of data must hold the information for 8 of the dots. Byte number 1 holds the information of the dot at the very top left of the screen and the 7 dots to its RIGHT. Byte number 2 holds the information for the 8 dots directly BELOW the ones held in byte number 1 and byte number 3 holds the 8 dots below that. However, when it reaches byte number 9, it doesn't take the 8 dots underneath those stored in byte number 8, that would be too easy, what it does is take the 8 dots to the RIGHT of those stored in byte number 1 and then byte number 10 starts moving downwards once again. When it reaches byte number 17, the information is again taken from the RIGHT, rather than from below and it is the 8 dots to the right of byte number 9 which are stored. So, as you can see, after every 8 bytes downwards the next byte is taken from the top of the screen again, but to the RIGHT. In hi-res mode, then, you have 40 bytes across the screen (a total of 40 bytes times 8 dots each equals 320 dots) by 200 bytes down the screen, which gives you the total of 8000 bytes (and if you multiply the 320 dots across the screen by the 200 bytes down the screen you return to your original figure of 64000). We'll call this 8000 bytes of data the PICTURE MEMORY.

In hi-res, the 1000 bytes of COLOUR MEMORY decides which colour the dots will be displayed in (makes sense really, doesn't it). Remember when I was explaining how the PICTURE MEMORY was stored I said that it stored 8 bytes downwards and then moved to the right to store the 9th byte. Well, one byte of COLOUR MEMORY decides the colour for ALL of those 8 bytes in the PICTURE MEMORY. A point of note is that it takes only half a byte of COLOUR MEMORY to decide which colour should be displayed, the other half of the byte is completely ignored.

As can be seen from the previous paragraph the dots have a choice of 2 colours, either the background (or OFF) colour, or the colour chosen from the COLOUR MEMORY (the ON colour). When it comes to MULTI-COLOUR hi-res graphics then that dot

has a choice of 4 colours, these are the 2 that it can normally choose from and an additional 2. Of course, it doesn't receive these for free, there are a few restrictions. The main restriction is that each dot becomes tied to it's neighbour. What I mean by this is that if you make dot number 1 blue, then dot number 2 must also be blue. If you remember, one byte of PICTURE MEMORY contains 8 dots. Before, you could control all 8 of these dots individually but were restricted to one colour and a background colour. Now, because every two dots are tied together, it seems as if you only have control over 4 dots in each byte but each of these dots has doubled in length and has a choice of 3 separate colours and a background colour. The 2 dots which are tied together are called a BIT PAIR so, in each byte of PICTURE MEMORY, there are 4 BIT PAIRS.

So each BIT PAIR has a choice of being displayed in one of 3 colours or the background colour, how does the computer decide which colour to use? I knew you'd ask that. Well, as you now know (hopefully), each BIT PAIR consists of 2 dots. If both of these dots are OFF, then both dots are displayed in the background colour. If the first dot is OFF and the other is ON then both dots are displayed in one of the other 3 colours. If the first dot is ON and the other is OFF then both dots are displayed in another of the 3 available colours. Finally, if both dots are ON then both dots are displayed in the final of the 3 colours.

All that remains is for me to explain where the 3 colours are stored. One is stored in the COLOUR MEMORY as I mentioned earlier. The other 2 are both stored in the same byte of the SCREEN MEMORY. This is possible because, if you remember, each colour choice only takes up half of a byte.

If you have followed this through then you should hopefully now have a reasonable understanding of how hi-res works on the Commodore.

As you probably know, your colour printer has 4 main colours, these are black, red, yellow and blue. If the picture on the screen contains any of these colours then printing them on to paper is obviously no problem. The problem occurs with the other colours. Ther eare 16 different colours that could appear in a picture and it is possible

for all of these to be displayed in the same picture. We need some way to differentiate between the colours when they are being printed onto paper. The way around this is to mix 'n match the 4 colours to make it look as if there are 16 different colours being printed. This is done as follows.

The black, red, blue and yellow presented no problems as explained earlier, however, to make them clearer, the computer went over any dots appearing in these colours twice.

The next easiest colours to display were purple, orange and green because the computer justs prints all dots in one colour and then goes over the top of these dots in another. For example, it printed all the green areas in yellow, and then went over the top in blue. Also, for the green, it then repeated this to give a more distinguished green but, for the orange and purple, colouring it once was enough.

The light red, light green, light blue and dark grey were printed by alternating between one dot of colour followed by leaving the next dot white. For example, the first line of light red was printed as follows, a red dot, a blank dot, a red dot, a blank dot etc and the second line was a blank dot, a red dot, a blank dot, a red dot etc. Line three was then a repeat of line one.

Cyan and brown was created by printing the first line in one colour and the second line in another. For example, the first line of cyan is actually in blue and the second line is in green.

Finally, the medium grey and the light grey were drawn as patterns. The first and third lines of medium grey were printed as follows, one dot black, one dot blank, one dot black and one dot blank etc, the second was one dot blank, one dot black then two dots blank etc, and the fourth was fully blank. Light grey had one dot of black on the first line and another dot of black on the third line and that was it.

As you can see, up to four different pieces of data are required in order to create some of the colours so it was easiest to make the patterns for all the colours take up four pieces of data, even though it meant that the four pieces for some of the colours were all the same.

Your Commodore is switched on, your colour printer is ready, CDU Paint (from CDU, March/April issue) is loaded and you've just created a graphics masterpiece that you want to show the world. Sound familiar?

So, what do you do in this situa-

FEBRUARY 1990 £2.75

Commodore DISK USER

THE 1st MILLION

The Strategist

Tomorrow's Tomorrow

Problems solved

ISSN 0953-0614

02

9 770953 061007

tion? Well, the world could enter your bedroom to see the picture in glorious technicolour on your T.V. screen, but the room may just become a little crowded. Alternatively, using the CDU Paint printer option, you could take a printout of the picture and show the world your creation in glorious technicolour and white.

Imagine having a program which could print out your artwork in colour, just as it appears on screen. That would be great, wouldn't it? Well, imagine no longer because Printpic is here and guess what, it prints out your artwork in colour, just as it appears on screen.

You're still not happy? Why not? Oh, I see. You, like me, can't even draw the curtains without making a dog's breakfast of it but you would still like to print out some pictures in colour. Fussy, aren't you. Okay, I'll tell you what I'll do. If you own the excellent Expert cartridge from Trilogic then I'll let you print out ANY picture that you want, even those of commercially release games. I hope that will keep you happy.

What? You want a program that will DRAW the pictures for you as well? Some folk are never happy...

Colour printing

Although there are only four main colours available on a colour printer (these being black, red, blue and yellow) it is possible, by using a variety of methods, to distinguish between the 16 different colours available in the 64.

These methods included alternating between a dot of colour and a blank dot to create a lighter colour, drawing a line in one colour followed by a line in another to seemingly 'merge' the two colours, going over the top of one colour with another to produce a third ('mixing' colours) and finally, creating patterns with varying amounts of white space for the lighter greys.

Some colours have to be gone over twice to darken them so, for each line of the picture to be printed on paper we require 8 lines of data, two in each of the 4 main colours (the second line in each colour being for those dots that require darkening).

Program routines explanation:

$8C00-$8C2A: The SCREEN MEMORY is usually at $0400, this routine changes it so that SCREEN MEMORY is temporarily at $8800 and character memory is at $9000. Finally it clears the screen.

$8C2D-$8C55: Data for screen messages.

$8C56: Temporary storage byte.

$8C58-$8C9F: Prints the screen messages onto the screen.

$8CA0-$8CC8: Waits for you to indicate your choice 'A' or 'B' and stores a different data byte in $8C56 depending on your choice. (The choice is between a CDU Paint picture file or a Trilogic file).

$8CC9-1$8D3D: If you chose a CDU Paint file then this routine firstly transfers the 8000 PICTURE MEMORY bytes to a new area of memory ($6000 onwards), then the 1000 SCREEN MEMORY bytes for the picture are transferred to $4000 onwards and finally the 1000 COLOUR MEMORY bytes are transferred to $4400 onwards.

$8D40-$8D8D: For a Trilogic file the 8000 bytes of picture memory are already at $6000 onwards so only the 1000 SCREEN MEMORY bytes and the 1000 COLOUR MEMORY bytes need to be transferred.

$8D8E-$8DD2: Data for another couple of screen messages.

$8DD8-$8DFA: Prints these new messages onto the screen.

$8DFF-$8E14: Data for a final screen message.

$8E18-$8E29: and prints this message to the screen.

$8E2A-$8E37: Waits for the spacebar to be pressed.

$8E38-$8E73: Displays the picture on the screen.

$8E74-$8E9C: If the F1 key is pressed then it changes the screen/border colour until the picture is displayed in the correct one. If the F7 key is pressed then this means that the screen/border colour is correct and it begins the actual printing process.

$8E9D-$8EFD: Sets a counter to 0 (if it reaches 40 then that means that 40 vertical character lines of picture have been printed or, in other words, the picture is finished printing). The rest of this memory is taken up with some routines to fix minor bugs that were spotted while testing the program.

$8EFE-$8F38: Moves a vertical line of picture data, which consists of 200 bytes, into a new memory location from $3000 onwards. The line of data is taken vertically rather than horizontally because the picture will be printed sideways onto the paper. Since the picture is to be printed sideways, then this vertical column of data is stored from the bottom up else the printout would be back to front. Since there are 40 vertical columns that the computer has to choose from and only one has to be transferred each time then the computer needs to be told which one to take. This is the job of the counter which was set up in the routine from $8E9D.

$8F39-$8F67: This moves a vertical column of SCREEN MEMORY into $3100 onwards. Since this vertical column contains only 25 bytes of data then each byte is repeated 8 times to make a total of 200 bytes of data which is the same as the number of PICTURE MEMORY bytes transferred.

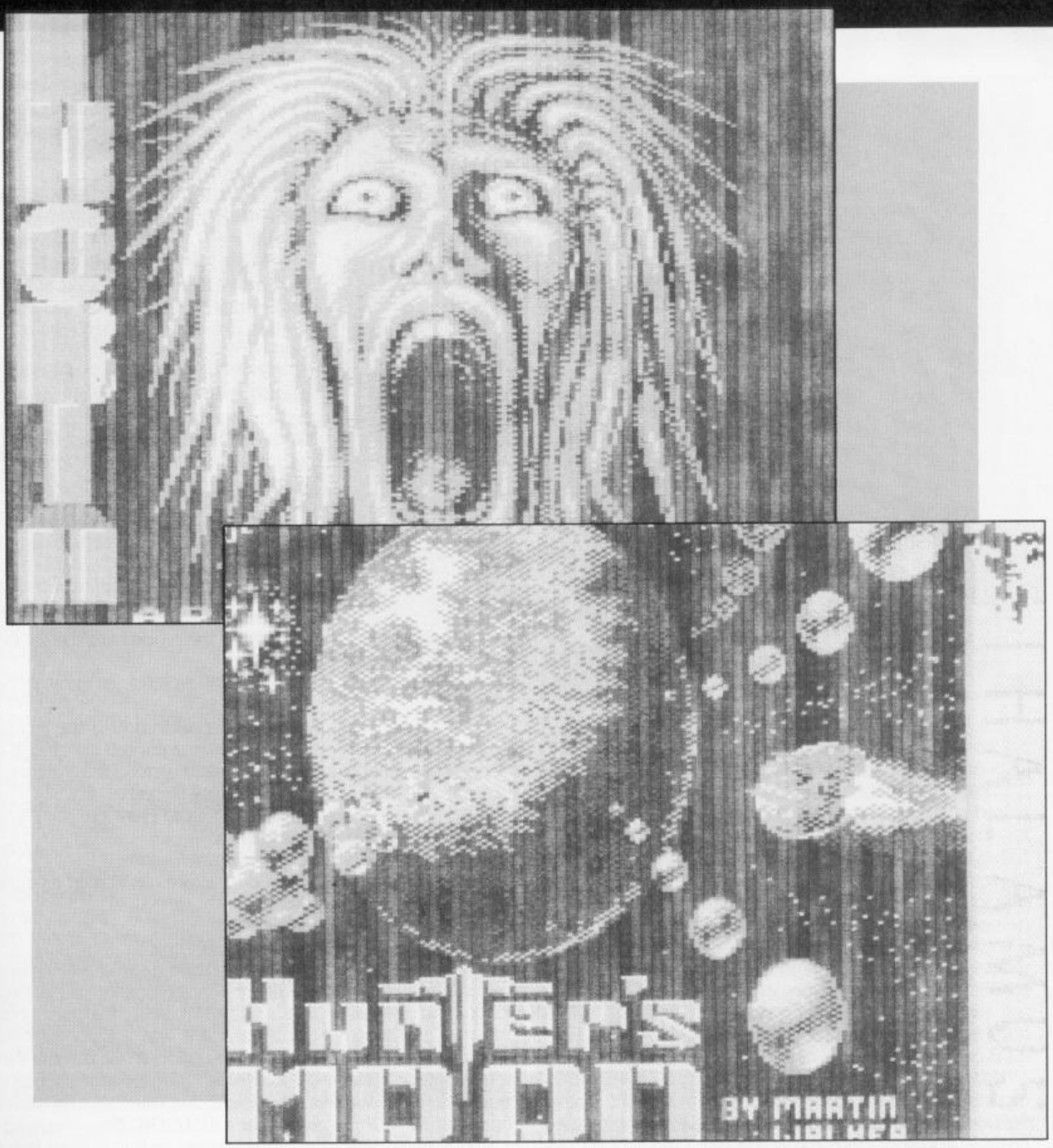
$8F60-$8F96: As above but transfers 25 bytes of COLOUR MEMORY into $3200, again repeating it 8 times.

$8F97-$8FC9: Each byte of SCREEN MEMORY actually represents 2 colours. This routine just splits up the 2 colours to make it easier later on. The first colour is stored back in $3100, the second is stored in $3300.

$8FCA-$8FFF: The actual printout will be twice as big as the picture on the screen (i.e. the screen is 320*200 dots, the picture will be 640*400 dots or else it would be too small on the paper). All this routine does is take all the data that we've stored in 200 byte chunks from $3000-$3FFF and doubles it into 400 byte chunks, storing it in $3400 onwards. Although this routine is possibly unnecessary it just makes the actual printing process easier because there is enough data manipulation going on without making it more complicated.

Also, all the above routines basically moved data here, there and everywhere in memory making changes as it went along. I found it a lot easier to do things in these steps rather than trying to make all the changes within one routine.

$9000-$9066: As I explained in the section headed 'Colour printing' there are four main colours. Some colours require each of the main colours printed twice to darken them while others require to be gone over only once. For this reason it is best to keep 2 lines of information for each of the



four main colours. The first line holds the data for ALL dots required to be printed in that colour, the second line just holds the data for the dots which need to be printed AGAIN in the same colour. This gives us 8 lines of information each comprising of 400 data bytes and this is stored from $1000-$2FFF. This routine just clears out that area of memory.

$9067-$9082: Sets up the data for the printing routine and stores it in the data area which begins at $9800.

$9083-$9093: Gets the next byte of the picture from $3000 onwards and temporarily stores it in $9858.

$9094-$909C: To know which colour to print we need to know what each BIT PAIR represents. The value of the BIT PAIR tells the computer what colour to print in. The colour is either the background (screen) colour, one of the colours from the SCREEN MEMORY or the colour from the COLOUR MEMORY.

If the first bit of the BIT PAIR is a '1' then the program jumps to $924A.

$90A6-$90AD: In a process of elimination by the above routine we now know that the BIT PAIR is equal to 00 and therefore represents the background colour. This colour is stored at $9851.

$90AE-$90C3: Each of the bytes of PICTURE MEMORY has 4 BIT PAIRS and, depending on which BIT PAIR we're currently dealing with, requires a slightly different routine to be used. This routine just finds out which BIT PAIR is being dealt with and jumps to the appropriate routine.

$90C4-$9134: This routine is used if we

are currently dealing with the first BIT PAIR of the byte. The colour which this BIT PAIR represents will have been stored in $9851 previously, so the data for this colour (see 'Colour printing') has to be taken from where it is stored in memory ($9900-) and the information split into the 8 lines of data needed (see explanation at $9000-). When this is finished it jumps back to $9094 ready to analyse the next BIT PAIR of the byte.

$9135-$918C: If we are dealing with the second BIT PAIR then this is the routine used to split the colour into it's 8 component lines of data.

**$918D-$91D5:**$ Routine if we are dealing with the third BIT PAIR.

$91D6-$9246: Routine for fourth BIT PAIR, in addition, since this is the last BIT PAIR of the byte, the end of this routines prepares for a new byte of the picture to be analysed rather than just a new BIT PAIR.

$924A-$9260: By the process of elimination (at $9094), the BIT PAIR must be equal to 01, so the colour must be one of the ones that were originally stored in the SCREEN MEMORY (which is now stored in 400 byte format at $3A00-). This routine then stores the correct colour in $9851 and jumps to $90AE.

$9261-$9269: Remember, the routine at $9094 jumped to here if the first bit of the BIT PAIR was equal to '1'. This routine checks to see if the second bit is eual to a 0 or a 1, if it is a 1 then it jumps to $9278.

$9279A-$9277: The BIT PASIR is therefore equal to 10 so the colour is the other one which was originally stored in the SCREEN MEMORY (and which is now stored in 400 byte format at $3600-).

$9278-$9285: The BIT PAIR is equal to 11 so the data is that originally stored in COLOUR MEMORY (and is now stored in 400 byte format at $3800-).

$9286-$93EC: By storing bytes of data straight into the code at $93ED we can cause the subroutine at $93ED to do slightly different things each time that it is called. In other words, it causes the subroutine to SELF MODIFY. This part of the code causes the subroutine to print the eight lines of data (stored at $1000-) in the appropriate one of the 4 main colours. It does this by first printing the two lines of YELLOW, then the two lines of RED, then BLUE and finally BLACK. Since the printout is twice the size of the screen (as explained in $8FCA), then what will be printed on paper will only be for HALF of the byte (or the data contained in the first two BIT PAIRS) so this routine then moves the paper up one line and, taking the data from $2000-, repeats the printing of the lines starting again at yellow but for the other half of the byte (or the data contained in the last two BIT PAIRS).

When it has done this it increases the counter defined at $8E9D and if it is not equal to 40 ($28) it jumps to $8EFE which repeats the whole process but for another vertical column of picture data. If the counter did equal 40 then it returns to BASIC ready for another picture or for switching off etc.

$93ED-$9463: The subroutine that self modifies. Basically this subroutine does the actual printing onto the paper.

$9800-$9CFF: The data needed for the program, which includes the four bytes of information required to creat the 16 colours.

Instructions:

Firstly, load the program in the following way:
LOAD "PRINTPIC",8,1, or load via the menu.

Next you have to load the picture which you want printed out. If it is a TRILOGIC file then load it as follows:
LOAD"(picture filename)",8,1
but if it is a CDU Paint file then load it as follows:
LOAD"(picture filename)",8
When the picture has loaded type the following:
SYS 34840
which runs 'printpic'.

You will then be asked to choose the picture data type that you loaded. If it was a TRILOGIC file then press key 'A' and if it was a CDU Paint file then choose option 'B'. Please note that the border will flash for about a second if you chose a CDU Paint file but it won't for a TRILOGIC file.

You will then be told to press space (which will result in the picture being shown on the screen). When you have the picture on the screen pressing the 'F1' key will circulate through the screen/border colours until you find the one which is correct for the picture. Note, each press of the key changes the colour, holding the key down does nothing.

When you are happy that the colour is right then press the 'F7' key and the printing will begin. That's all there is to it but now your favourite pictures are about to become permanent additions to your bedroom wall.

FREE ISSUES

A subscription to your favourite magazine is the best way of making sure you never miss an issue.

And from now until 28th February 1990 you can get extra copies ABSOLUTELY FREE, by taking advantage of our special Christmas subscription offer. With a monthly title for example, this means you get 15 issues for the usual price of 12.

Order your subscription today using the coupon below and you will receive the best in reading entertainment right into the 1990's. This offer is also open to subscribers wishing to extend/renew their current subscriptions.

Standard subscription rates ▶ ▶ ▶

	UK	Europe	Middle East	Far East	Rest of World
Monthly titles (15 for the price of 12)					
A & B Computing	£21.00	£30.30	£30.60	£34.10	£31.25
Aeromodeller	£23.40	£28.20	£28.40	£30.20	£28.70
Antique Clocks	£27.00	£32.40	£32.60	£34.70	£33.00
Citizens Band	£18.00	£21.95	£22.10	£23.60	£22.35
Commodore Disk User	£33.00	£39.00	£39.30	£41.60	£39.70
Electronics Today International	£18.00	£22.20	£22.40	£24.00	£22.70
Film Monthly	£16.80	£21.10	£21.30	£22.90	£21.60
Ham Radio Today	£18.00	£22.50	£22.70	£24.40	£23.00
Micro Music	£21.00	£28.00	£28.30	£30.90	£28.80
Military Modelling	£18.00	£24.80	£25.10	£27.65	£25.50
Model Boats	£18.00	£22.40	£22.50	£24.20	£22.80
Model Railways	£16.80	£22.70	£22.90	£25.10	£23.30
Photography	£23.40	£30.50	£30.75	£33.45	£31.25
Popular Crafts	£18.00	£23.30	£23.50	£25.40	£23.80
Radio Control Model Cars	£17.40	£22.30	£22.50	£24.35	£22.80
RCM&E	£16.80	£22.80	£23.00	£25.30	£23.40
Radio Modeller	£16.80	£22.40	£22.60	£24.80	£23.00
Renovate	£18.00	£22.80	£23.00	£24.80	£23.30
Scale Models International	£18.00	£22.60	£22.80	£24.55	£23.10
Video Today	£18.00	£23.20	£23.40	£25.35	£23.70
Video X	£18.00	£22.30	£22.50	£24.10	£22.80
Which Video?	£18.00	£22.35	£22.50	£24.15	£22.80
Woodworker	£18.00	£25.20	£25.40	£28.10	£25.90
Your Amiga	£23.40	£29.00	£29.20	£31.40	£29.60
YC - Your Commodore	£23.40	£31.00	£31.30	£34.20	£31.80
Alternate monthly titles (8 for the price of 6)					
Radio Control Boat Modeller	£8.70	£11.30	£11.40	£12.35	£11.55
Radio Control Scale Aircraft	£13.50	£16.65	£16.80	£17.95	£17.00
Practical Wargamer	£11.70	£14.80	£14.90	£16.10	£15.10
Fortnightly title (28 for the price of 24)					
Model Engineer	£31.20	£40.20	£40.50	£43.90	£41.10

Please commence my subscription to ..... with the ..... issue.

I enclose my cheque/money order for £..... made payable to ARGUS SPECIALIST PUBLICATIONS

or debit my Access/Barclaycard number ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐ Card expiry date .....

Signature ..... Name .....

Address .....

..... Postcode .....

Please return this coupon with your remittance to:
Infonet Ltd. (CDFI/3) 5 River Park Estate, Billet Lane, BERKHAMSTED, Herts. HP4 1HL UNITED KINGDOM.

ON THE DISK

Base-Ed

ase-Ed

se-Ed 2

Base-Ed 2

This version of Base-Ed has been modified so that the 'Batch Processing' option now functions correctly. Many of you will have experienced some difficulties on the original due to an error in the compiler. There is also an enhancement to the 'Interrogation' module which allows you more flexibility over the conditions you apply to the search.

Base-Ed 2 is an update of Base- Ed originally published May/ June CDU with a few enhancements/modifications.

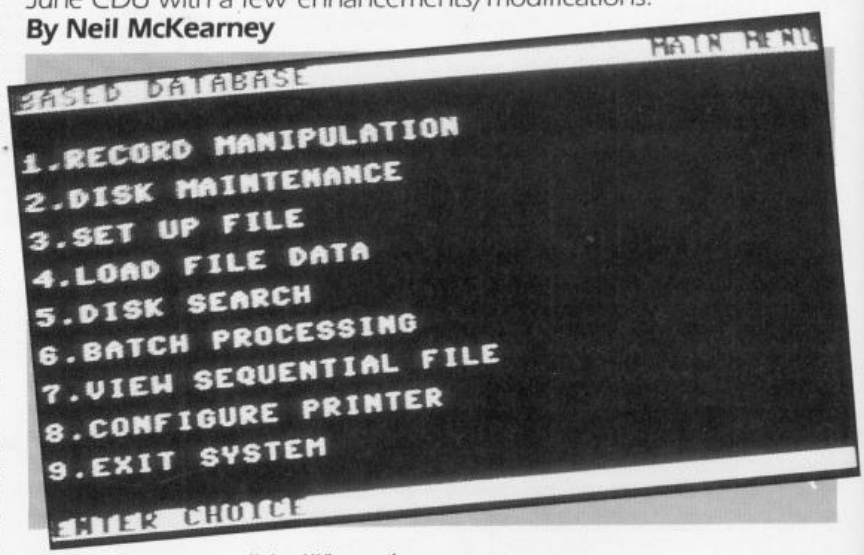
By Neil McKearney

Base-Ed is a random access database allowing a maximum of 500 records per disk which may be entered and then subsequently viewed, rectified, deleted and interrogated. Each record can have a maximum of 39 fields but the record length must not exceed 255 characters.

Setting Up a File

Select the *'Set up file'* option from the main menu. You will be asked for the name of your file. Enter anything you like as long as it includes no punctuation and does not exceed 16 characters. After this, the program asks for the number of fields. Enter the amount, which must be less than 30, and you will be prompted to enter the field names and their lengths. Note that all the lengths added must make no more than 255 or the fields will have to be entered again. The program then asks if all the data is correct. If it is type 'Y' to proceed, otherwise type 'N' to re-enter the data.

A message will be displayed on the screen to place a disk in the drive. Make sure there is no valuable data or code on the disk because it will be formatted. Press return and the program will now prepare the disk for use in your file. A flashing box in the top left corner will indicate when the program is working on your disk. When the process has finished and the message to press a key comes on the screen, press any key to return to the main menu.



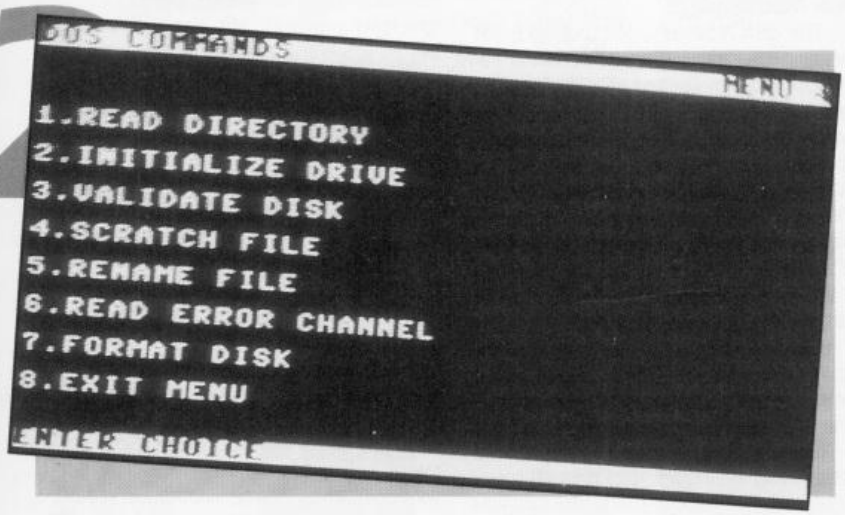
Recording Manipulation

This is the main part of the program where all your work will be done. The options allow you to enter, amend, delete, read and print records or to interrogate the file and search the disk.

Enter Record

Enter the number of the record in the file and when you press return, the program will display the record number and the track sector to which the data will be written. Now you can enter the record.

Type in the data for each field, pressing return after each. When you have finished entering the record the program will ask if it is all correct. Make your decision and press 'Y' or 'N'. The



program will then return to the *'Which record'* prompt. To leave the record entry section press RETURN on this question.

Amend Record

This is the first part of the program in which you can use the INDEX which allows you to call a record by the first field in the record. If, for example, you choose to use a file in which the first field was *Name* you could type in the word 'Name' to view it. If you are not using the index press return, when asked for the field press return again and you will be asked for the record number.

The amendment process involves calling up a record, viewing it and then deciding whether or not to alter it. Call up the record using INDEX or NUMBER. Note that the number method can still be used even if you are using an index. Then, when asked if you want to change it, type 'Y' or 'N' accordingly. If you typed 'Y', the program will ask you to enter the record and, if you have entered all the data correctly, the record will be written to the file and the amendment made to the index. If you typed 'N' the program will return to the record manipulation menu.

Delte Record

Delete allows one record or a group of records from the file to be erased. On selecting this option you are greeted with a sub-menu.

Select the option you require using the keys 1, 2, 3. If 3 was pressed then you will return to the manipulation menu. On pressing 1, you must enter a record number, or first field name if you are using an index. To enter a number, type the number and hit return, otherwise hit return to enter a field. If you have made a mistake and do not want to enter a field here then type return to go back to the sub-menu. If you entered a field or number then, the program will delete the record correspondingly. The program then returns to the sub-menu.

If Option 2 is selected from the sub-menu, two numbers must be entered. These numbers are the record to start deleting from and the record to end deleting. Each record will be deleted and the program will return to the sub-menu. To abort this option, type zero for one of the required numbers.

Reading Record

Using another sub-menu, this option allows you to read one record or a group of records. It functions in exactly the same manner as the *'Delete record'* option. Refer to the previous heading for details.

Interrogate File

1. Select the interrogation from the record manipulation menu as usual.

2. The prompt 'Enter the field number' appears on the screen.

3. Enter a number which corresponds to one of the fields in your file which has been previously listed on the screen.

4. The next prompt is 'Enter interrogation string'.

5. Your entry need not be a string, it can be a number. This is the entry or which the interrogation will be formed for that particular string.

6. The final prompt is 'Enter interrogation type'. 'There are seven types of interrogation type in Base-Ed 2and they are as follows:

1) > – The field is greater than the string/number specified in the interrogation string.

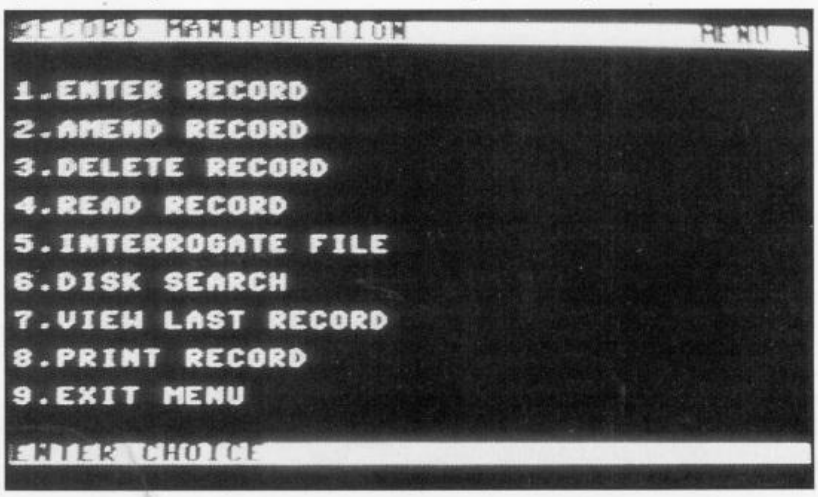
2) < – The field is less than the string/number specified in the interrogation field.

3) > = – The field is greater than or equal to the string/number specified in the interrogation string.

4) < = – The field is less than or equal to the string/number specified in the interrogation string.

5) = – The field is equal to the string/number specified in the interrogation string.

6) < > – The field is not equal to the string/number specified in the interro-



gation string.
7) MID$ – If this option is selected, the program will search through the specified field for any occurrence of the field specified in the interrogation string.

Your entry to the last prompt should be a numeric one in the range 1-7. You will then be asked if you want to enter another interrogation. Enter 'Y' or 'N' accordingly. Up to 30 interrogations can be done at one time.

Disk Search

This part of the program will search the disk records for any string which you enter. The first two questions again ask for the record to start and end searching, enter nonsense data to leave this option. The next question asks you to enter the piece of data you wish to be found and the disk will then be searched. When, and if, the string is found the program will display the record at which it was found and the track and sector of that record. Type any key to return to the *'Record manipulation'* menu. If the string was not found then hit any key when the search has finished.

View Last Record

Using this option simply allows you to view the contents of the last record entered, its length and the track and sector it was written to.

Print Record

Through another sub-menu, this option allows you to print one record or a group of records. It follows exactly the same format as the *'Delete record'* option.

Exit

Typing 9 at the *'Record manipulation'* menu will return to the main menu.

The Other Features

Base-Ed 2 is designed to work with a multiple drive system and will therefore direct all disk interaction to the device from which it was loaded. It recognises device numbers from 8 to 11.

Base-Ed 2 also has an automatic keybeep option which is turned on at initialisation. From then on, F1 will switch the beep off and F3 will switch it back on.

Disk Maintenance

The disk maintenance section is for sending disk operating commands for the purpose of updating the disk and viewing its contents. If an error occurs which is connected with the disk drive, go to option number 6 in the *'Disk maintenance'* menu and check to see the error. If necessary, refer to the disk driver user manual for an explanation of the fault.

Batch Processing

Batch processing allows you to set up a temporary file, manipulate it and, when finished, write it to the disk in one batch.

The first thing you must do in batch processing is to set the record specifications. This is how many records you will use and what their record numbers are. Select option 8 from the *'Batch processing'* menu. You are asked for start and end parameters.

1. Enter record to start processing.
2. Enter record to end processing.

Type two numbers within the range 1-500. Now you can manipulate your file by using the record entry, record amending, record reading and sort file options. The first three work in the same manner as the equivalent record manipulation options. *'Sort file'* sorts your temporary file into alphabetical order using the first field and the *'Save/Load sequential file'* option allows you to store or retrieve your temporary file for latter use or updating.

The *'Write to Random File'* option takes your temporary file and writes it on to your data disk at the appropriate places and updates your file index accordingly. 'Load from Random File' loads all the records which have been specified in the *'Batch Specifications'* option and puts them into a temporary file for manipulation.

View Sequential file

When asked whether you wish to view any unprotected sequential file on the screen or printer, type 'S' or 'P' accordingly. Then type the filename and, as long as it is legitimate, the file will be displayed. If you want to leave viewing the file at any stage hit the left arrow key. Press any key when the program finishes displaying the file.

Printer Configuration

To control the device number and print type of the current printer, use the keys 1, 2 3 to change the device number, double width printing and reversed printing. Hit the 'up arrow' key to leave this option. Hitting the 'M' key will bring you to the Mail label menu.

Mail Labels

Using the mail labels option you can create, store and use a format by which your labels will be printed. In the first heading you must enter the number of lines per label and the number of lines between labels. For every printed line you must then enter the number of fields on that line and then the numbers of the fields on that line. (The number of the field is simply where the field comes in the field order. So if you were using Name, Age and Address as your fields, Name would be 1, Age would be 2 and Address would be 3.)

Once this process is finished you have created as many formats as you like because there are SAVE and LOAD options for storing or retrieving a format.

When you want to print labels, select the *'Print Labels'* option and follow the prompts. Then enter the two numbers to start and end printing and the program will print labels in the current format. If at any stage you wish to pause printing type 'P', type 'C' to continue and 'E' to stop.

If you select option 9 from the main menu you will firstly be asked *'Are you sure?'* Enter 'Y' or 'N'. Then you will be asked *'Do you want to save the index?'* Again, enter 'Y' or 'N'. If 'Y' was selected then the index will be saved onto the disk. In both cases the system will be reset, leaving you at the power up screen.

Please note that it may be necessary to use a separate disk for the index. This is only the case if your first field involves a lot of characters and the file is relatively full.

Loading Base-Ed 2

Type LOAD "BASE-ED 2",8,1 followed by RUN. The progam will install itself automatically or select from menu.

These instructions merely summarise the functions and get you familiar with Base-Ed 2. Setting up and using your own file is the best way to learn about the system but remember to use unimportant disks for experimenting. We hope that Base-Ed 2 will help you with your filing needs.

The First Million

Would you like to become a high flyer? A modern day Yuppie? Don your flying suit, grab your filofax and earn yourself a million

By Darren Cook

Everyone at some time in their life has dreamed of becoming a millionaire. This game enables you to see if you have the natural abilities to MAKE money FROM money.

You begin the game with one thousand pounds. With this you can buy and sell shares. To find out the share prices you should examine the Financial paper. (Press 3 on the main menu). To purchase shares you press 1 on the main menu and type the number of the company you wish to buy (1-18), then you say how many shares you wish to purchase. The maximum number of shares available to you at one turn is 999. After each turn the shares value will alter. If you wish to wait until your shares change their value then press F3 (this waits for 10 turns) or press F1 (this waits just 1 turn).

To sell shares you press 2 on the main menu. A list of the shares that you own will be displayed. You then type in the number of the company you wish to sell and how many shares you want to dispose of.

If you have financial security of two thousand pounds then the bank will allow you to have a loan between one thousand and twenty thousand pounds. (Due to the high risk of no return). You must pay 16 per cent interest on a loan and you pay back over several turns. You are only allowed 1 loan at a time.

Once you have sufficient funds you are permitted to purchase and sell Gold. Press 8 or 9 on the main menu for this feature. You can only buy between 1Kg and 9Kg of Gold in each turn. The price of Gold fluctuates at the end of each turn.

Once you have Ten thousand pounds you may purchase a mining plot to enable the digging of Gold. To choose the best plot you can take out research in three different areas. Each area comprises of five different plots. Research costs three thousand pounds an area and five percentage ratings are shown.

Option 7 allows you to dig 600ft at a time. This will cost you five thousand pounds for the first 600ft. As you dig deeper, so the price increases. You can only dig to a depth of 6600ft. You are permitted one mine at a time only if your mine becomes exhausted you must close it down before you can buy another. This will set you back fifty thousand pounds.

There are two modes of gameplay, namely Random and Realistic. The random mode is for lucky players as the prices vary randomly each time. The more realistic mode has several companies which are always successful. Obviously, the more you play the easier it will be to work out which company will or will not do well.

Answers to all yes/no questions are obtained by pressing 'Y' or 'N' accordingly. All options are chosen from the main menu by pressing the corresponding key. Once you have made over one million pounds you have completed the game. Good luck and happy spending!

Being artistic on the C64 is not an easy thing. Here are a few tips that may help to overcome this

By Bizzmo

The graphic capabilities of the 64 are, to say the least, very primitive. With a resolution of 320 by 200 pixels in high-resolution mode. Let's face it, it's no Amiga. (The resolution most pictures are drawn in on the Amiga (Lo-Res) is just the same as the Hi-Res mode on the 64, but the greater range of colours available, and more noticeably the ability to put them right next to each other, makes the resolution seem much higher).

Things ger worse. To get more than two colours in an eight by eight pixel square, you have to enter multi-colour mode. This gives you four colours in each eight by eight square. However one of these must be the background colour. That leaves you with three. There is a drawback with multi-colour mode. It reduces the horizontal resolution by half, from 320 pixels to 160. That means the pixels are big.

So, how are you meant to get anything barely pleasing from your 64? Well, the first thing to do is to get a good art package. Producing a picture by POKEing memory locations would take forever and be exceedingly difficult, if not impossible! The best thing that I did was to get my hands on the Neos mouse and Artist 64 from Wigmore House. Trying to draw free hand using a joystick is a killer. Another good graphics package is the Advanced Art Studio for Rainbird.

Planning the Picture

This is more important on the 64 than on the 16 bit machines, due to the colour limitations. Trying to cram too many colours into too small a space is painful to say the least. The size of the subject in relation to the screen is also important. Because of the resolution, drawing anything small is going to be hard. Getting it to look reasonably life-like is nigh impossible.

Once you have decided what you are going to draw, it is then a matter of getting a representation of it onto the screen. I've found that the best way to do this is to use the LINE option. Trying to draw freehand is very tricky.

To get the picture on the screen, use the LINE option to draw the basic outline. It doesn't matter if it is a bit off, just use it to get the proportions right. This is the best way I've found, but some people may prefer to draw a small piece of the picture at a time and gradually expand it as they go, finishing each bit at a time. Using lines means that you can easily, and quickly, visualize the final product. To produce curves, just use a few small lines, at an angle to each other. The more lines you use the smoother the circle. Many objects have some symmetry, that means that one part can be mirrored to produce another part, this helps to get an object to look balanced. For example it is only necessary to draw half a face, as it can be mirrored to produce the other side. After achieving a rough outline, with the proportions roughly corect, you will need to add some detail, and some colour.

Adding Colour to the Picture

The 64 has a very limited colour-pallette. In all you've got sixteen colours to play with, so because of this limitation, shading becomes more important. To get shades 'in-between' the colours available, you have to use various patterns to 'mix' the available colours until you get the shade you want. The main way of doing this is to use alternate dot shading, where the colours are arranged in a checker-board pattern. This looks best if you stand at the other end of the room and squint. Obviously this method works better with some colours than others. Contrasting colours will work less well than colours of a similar shade. For example Red and Blue look awful, but Pink and Orange go well together. On the 64 the main colours Red, Green and Blue have a lighter shade, and there are three Greys which come in very handy when shading. Try not to have large areas of the same alternate dot shading. On many art packages there are spray options available, these can be a little messy if you are not careful, but can, with a bit of practice, become a very useful tool when shading. To colour the main areas of the picture use the fill option which will flood an area with colour. Be careful that the areas you want to fill are completely enclosed, or you may end up filling the entire screen! Once you have got the main areas coloured, use some alternate dot shading to 'blend' the colours and smooth out the hard edges.

Adding detail

Once you have got something that looks roughly like what you want it to, it's time to add some detail. The best way to do this is to use the ZOOM option to enlarge a part and edit it pixel by pixel. Before entering ZOOM, use either the line option, or freehand to mark where the various detailed parts are to go. Then use ZOOM to enlarge these areas. The most important thing to remember when you are in zoom mode is that you can't have more than four colours in the same character square, and one of these colours must be the background colour. Adding the detail is the most time consuming part of creating a picture, and it's normally best to break it up a bit, taking a break now and again. Zoom is also useful for tidying up those straggly ends and the odd stray pixel.

Anti-Alias

Anti-Alias or 'getting rid of the jaggies' as it is know in the trade, is probably the most useful skill to master, especially when the pixels are so large on the 64.

When you draw on a computer, anything other than horizontal or vertical lines produces jagged lines. This is called Aliasing. What makes things worse is that on the 64 in multi-coloured mode, these jagged lines are magnified. These jagged lines show up the most when you place a bright colour next to a dark one. So white on black produces the worst results. The way to get round this is to put an 'in-between colour' along the border of the contrasting colours. This reduces the contrast and so the jaggedness. Things can be improved still if two 'in-between' colours are used, as this smoothes even more, the transition from one colour to another. Thats all there is to anti-aliasing. Just this one technique greatly improves the quality of your picture. The best way to get to grips with graphics is to experiment, trying different colour combinations and shading effect.

That just about wraps it up for this issue. I will be imparting some more of my humble knowledge to you in due course.

FM-Dos

A short program to enhance the capabilities of your 1541 disk drive can make life so much better.

By Fergal Moane

When Microsoft designed Basic 2 they seem to have ignored a very important peripheral: the disk drive. The drives were an afterthought – slow, noisy and little software support. When the C128 came out, Commodore improved on all these points with Basic 7 and the 1571. With the 4K chunk at $C000 and a little bit of system software, FM-Dos attempts to create the Basic 7 disk handling commands – even the improved speed!

1) FM-Dos uses all the chunk of memory from $C000 to $D000 for its commands. The code to handle the extended basic is at $9E00,, with the top of basic lowered to protect it. Make sure that variables or code do not corrupt FM-Dos or you can say bye-bye to your program.

2)FM-Dos was really designed for the Basic Programmer who dabbles in graphics or machine-code. As $C000-$D000 and most zero page locations are occupied, it will seem less attractive to the machine-code programmer, but the turbo may still be useful. Note that interrupts are not used – programs using them will work correctly.

An alphabetic summary of the commands follows. Note that some commands have duplicates – these are for convenience and function identically.

BLOCK LOAD
Syntax: BLOAD, "filename" ,8,0,address
Loads a block of memory from disk. Address should be an address from 0-65535 where you wish the data to be loaded into. This command sets Basic pointers correctly, unlike LOAD "filename",8,1. The machine-code should be relocatable if you specify a load address.

BLOCK SAVE
Syntax: BSAVE, start, end, "filename",8
Saves the block of memory from start to end to disk. The start and end addresses should be greater than each other. Add one to the end address that you want to save for the last byte to be saved properly.

BOOT
Syntax: BOOT "filename"
Loads a Basic program from the current disk and automatically executes it. Machine code that is called by a Basic line will function correctly. BOOT "*" will load and run the first program on the disk. It is the approximate equivalent of pressing SHIFT/RUN STOP.

DISK APPEND
Syntax: DAPPEND "filename"
Appends the Basic program "filename" to the program currently in memory. The line numbers should not conflict in the two programs or problems could occur. Use the LIST command to check before the append. Note that this is a memory append and that nothing is appended on the disk. It should not be confused with the append mode used for sequential files in CBM DOS.

DISK CLOSE
Syntax: DCLOSE
DCLOSE terminates all communication with the disk drive and returns I/O to normal. It should be used at the start or end of a program which deals with disks. You should not use this command to close individual files after writing – they should be closed with the CLOSE command.

DISK ERROR STATUS
Syntax: DERR
This will output the disk status error message at the current print position. It is a fast command and so should be used after every disk access without much slowing effect. Use the list of error messages in the manual to determine the cause of the error.

DISK DIRECTORY
Syntax: DIR
DIR reads the disk directory and prints it to the screen. No memory is corrupted by this directory command, and control will return to your program after it has been read.

DIRECTORY
Syntax: DIRECTORY
DIRECTORY is merely the expanded form of DIR. See above for an explanation.

DISK COMMAND
Syntax: DISK "command"
Opens a channel and sends a command to the internal DOS in the drive. It replaces OPEN15,8,15 with something more memorable. See your disk drive manual for the multitude of commands that DISK opens up. DISK without a command will display the disk status. DISK$ will display the disk directory.

DISK LIST
Syntax: DLIST "filename"
DLIST is potentially the most usefulcommand as it lists a Basic program without actually loading it into memory. Listings can be examined with another program in memory without corruption. This has many uses and lines can be grabbed from a program by DLISTing it, pressing RUN/STOP to stop the listing and pressing RETURN over the required lines. Note that this will only work for Basic programs as it de-tokenises the text. Use SLIST for other filetypes. DLIST"$" lists the disk directory.

DISK LOAD
Syntax: DLOAD "filename"
DLOAD eliminates the need for obscure ,8 suffixes. It functions in the same way as the ordinary load command except the device defaults to disk instead of tape. Note that a filename is always required, even if it is only "*".

DISK OPERATING SYSTEM COMMAND
Syntax: DOS "command"
Accesses the disk operating system in the disk drive. It functions in the same way as the DISK command.

DISK SAVE
Syntax: DSAVE "filename"
Saves a program to disk. It operates in the same way as the ordinary save command, except no 8 is necessary. If you want to replace a file, use the syntax DSAVE"@0:filename". See REPLACE for an alternative method.

DISK VERIFY
Syntax: DVERIFY "filename"
Verifies an existing program on disk against the one currently in memory. This is useful to check for errors after a DSAVE, but errors with disks are very uncommon.

FUNCTION KEYS
Syntax: KEY
KEY initialises the FM-Dos interpreter

and sets up the functions keys for the following definitions:

F1 – DERR
F3 – DIR
F5 – TURBO
F7 – TURBO OFF

If these settings clash with your programs, RUN/STOP and RESTORE will switch off the functions keys, but not FM-Dos. KEY will then restore the function key afterwards. The function keys do not use interrupts and should be completely compatible with any program that alters the HIV. F3 is useful as you can call up a directory and cursor up to the filename and type DLOAD. This saves you having to remember the correct filename.

QUIT FM-DOS
Syntax: QUIT
Exits FM-Dos leaving programs and memory intact, but the interpreter inaccessible. SYS 40908 will re-enter FM-Dos. If you want to execute other programs, you are advised to switch your computer off and on again to clear FM-Dos from the memory.

REPLACE FILE
Syntax: REPLACE "filename"
Replaces the Basic file specified with the new program currently in memory. This is useful in updating programs. REPLACE does not trash disks, as it uses the safe form of save with replace.

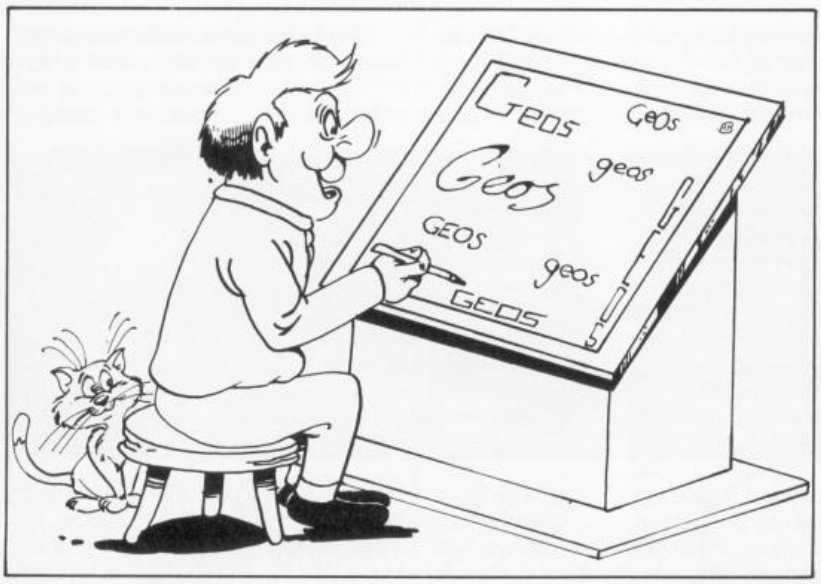
TURBO OFF
Syntax: OFF
This exits the FM-Dos disk turbo and returns to normal speed I/O. A message is given to confirm this. This is useful if the FM-Dos turbo clashes with your own programs, or a commercial protection scheme. See the TURBO command for details of the FM-Dos turbo.

TURBO
Syntax: TURBO
TURBO enables the FM-Dos turbo. It loads at approximately 10 times the normal speed at around 4K per second. This equals the 1571 in its top Burst Mode transfer speed. Any commands possible (DLOAD, BOOT, BLOAD etc.) will attempt to use FM-Dos turbo if it is enabled. This results in greatly descreased loading times for all these commands. It will work with all PRG files, Basic or Machine Code. Remember not to load programs that will overwrite FM-Dos or a crash will occur. The screen is blanked during loading to increase speed. FM-Dos turbo should be disabled with OFF or F7 if conflicts with other loading systems are likely to occur.

DISPLAY FILE
Syntax: TYPE "filename,filetype"
TYPE is roughly equivalent to the MS-DOS TYPE command. It will attempt to display a file onto the screen. All characters are printed, so some files may look very messy. This is because of control codes. Its main use will be the displaying of sequential wordprocessor files. Use the letters below for filetypes:

P – Program (PRG)
S – Sequential (SEQ)
L – Relative (REL)
U – User (USR)

Hopefully, FM-Dos will help you maximise the speed and versatility of your disk drive.



Geos Fonts

Another 4 fonts are provided in this month's issue of CDU for users of the GEOS package.

By Brian Sedgebear

In Jaunary's edition of CDU we provided the first 4 of 12 new fonts for users of the GEOS package. This issue sees the next 4. (Western, Scripture, Dingbats and Feathers).

Now as promised I will go into a little more detail on Geos, dealing in this issue with how character style, font control, font size and also font file structure operate within Geos. Even if you do not use Geos you may find this method of font control interesting for any of your software.

In GEOS with the exception of BSW9, which is memory resident and often referred to as the system font, the fonts are loaded individually from disk when required, and are stored in the application memory from $0400 to $5fff. A font in GEOS terms is a group of character sets of different point sizes but in the same letter format. A character set in turn is a group of 96 individual characters (97 in the case of BSW but don't ask me why!) all in the same point size or height but they can be different widths to allow proportional print. The point size is the height in pixels of the character and is 1/80th of an inch. Geos identifies each of these fonts and characters sets by its own unique font identity number (Font ID). The font ID is 10 bits long, bits 0-5 forming the point size and bits 6-15 forming the font number. This font ID must never be the same for any two character sets and fonts or GEOS will not know where to look for the relevant character data. GEOS embeds this code in the text file in the text buffer of the application at the appropriate point for changing font.

GEOS also has the capacity to use different styles like bold, underline and outline with any font. GEOS does this in a similar way as font changes, by embedding the correct control code for a style change in the applications text buffer at the point where the change must take place. It is important to know that GEOS does not need a new character set to produce a new style but, as in the case of bold text simply makes each character one pixel wider when it is sent to the screen or printer. The same technique is used to produce slanted italic characters and so on. Below, I have gathered some of the control codes used in a raw GEOS text file.

ASCII CODE	ACTION
14	UNDERLINE ON.
15	UNDERLINE OFF.
18	REVERSE ON.
19	REVERSE OFF.
24	BOLD ON.
25	ITALIC ON.
26	OUTLINE ON..
27	RETURN TO PLAIN TEXT

GEOS font files are stored as VLIR (variable length indexed record) files with a GEOS file type of FONT. Each VLIR file uses an indexed block to point to a total of 127 individual records, each of which can be unused, or of variable length.

Each record can therefore be used to store one character set within the font. Geos works on the basis that in font files record one will contain the one point character set, record two will contain the two point character set and so on. If the pointer is set to $00,FF then no character set in that size exists in this font. The font directory entry is formed as follows.

BYTE.	REMARKS.
2	File Type.
3-4	Index Track and Sector.
5-20	Filename.
21-22	Header Block Track and Sector.
25-29	Date and Time (Y,M,D,H,MIN in 24 hour format)

Looking at bytes 21-22 in the font directory entry above we can find the location of the header block. Geos uses this block for general information about the font including the font ID codes, icon data, author's name, desktop info and so on. The header block format for a font file is as follows.

BYTE.	REMARKS.
2-4	Icon dimensions.
5-67	Icon data.
77-92	Fontname for desktop info
96-127	use. Size in bytes of each character set from smallest to largest padded with 0's.
128-129	Unique font ID code.
130-159	Font ID for each character set, from smallest to largest padded with 0's.

Looking at bytes 3-4 of the font directory entry we can see the location of the font INDEX block. This simply tells GEOS the start of each record, or in the case of fonts the start of each character set from smallest to largest. Finally we come tothe first block of each record (or character set) which gives GEOS specific information about the particular character set that follows. This block is laid out as follows.

BYTE.	REMARKS.
0	Underline position on character.
1-2	No. of bytes long.
3	Font size.
4	Pointer to start of character width table.
6	Pointer to character data.

Usually from about 8 to about 200 we will see the character width data which adds the width of the latest character onto the previous total.

Usually following this will be the character data.

Well that will give you a good start, I hope in the operation of how GEOS fonts work and using a disk monitor you will be able to look at how one of the GEOS fonts is constructed.

Input Drivers

Now we will take a quick look at Input Drivers in the GEOS operating system.

Berkeley Softworks have provided us with an impressive array of input drivers enabling anything from a lightpen to a mere joystick to be a flexible input tool. The input driver file is used to form the link between the input device and the GEOS operating system. It is vital to provide this link so that almost any input driver can be accommodated by GEOS without changing the operating system, but by simply loading a new input driver. GEOS loads the input driver file nearest the beginning of the desktop into an area of memory starting at $FE80. A new file can be loaded to overwrite the old one at any time the user requires a new input device. The input driver is divided into three main routines which GEOS can call, the first with a vector at $FE80 is a routine called when the driver is loaded to initialize any local and globe variables used. It can also be used to initialize the actual input device. The second routine pointed to by a vector at $FE83 is used to slow the mouse down and is called whenever the user clicks on a menu or icon. It basically reduces the speed and or acceleration so that the pointer does not skate off the edge of the menu when the input device is moved slightly. The final routine has its vector at $FE86 and is the control routine and is called at each interrupt to update the mouse position and check the status of any input buttons. GEOS has global variables which are used to control the position of the pointer ect, the four most important variables are the two byte (word) used for the Xposition of the pointer, the Yposition of the pointer, the button flag and the control flag which we will look at next. There are also some other variables which are mentioned at the end of this article with a brief description of their use. The button flag (also called the mouse data flag) is set to a negative value if the fire button is released and to a positive value if the fire button is pressed. The control flag (also called the pressflag) has two bits of importance, bit 5 (mouse bit) is set when there is any change in the button position and bit 6 (input bit) which is set if there is any change in the input device since the last interrupt. The routine for slowing the mouse when on menus is not usually required when devices such as mice and light pens are used.

There are some additional global variables which are very useful which can be loaded with values during the drivers initialization routine or changed by the preference manager. These variables are particularly valuable for input devices which are not proportional, like the common joystick. These variables are the maximum mouse speed, the minimum mouse speed and the mouse acceleration which is used to control how fast the mouse moves to its maximum speed. The input driver routine must make use of these variables when calculating the mouse X and Y positions.

Mouse X Position	$003A-3B	
Mouse Y Position	$003C	
Press Flag	$8505	
Mouse Data	$8505	
MAX Mouse Speed	$8502	
MIN Mouse Speed	$8502	
Mouse Acceleration	$8503	
Mouse Top	$84B8	Uppermost vertical position of pointer.
Mouse Bottom	$84B9	Lowest vertical position of pointer.
Mouse Left	$84BA-BB	Furthest left pointer can go.
Mouse Right	$84Bc-BD	Furthest right pointer can go.

The address for the Input Driver variables is given in the table below.

The driver need know nothing about the final four variables, I included them to show you that you need not limit the X and Y positional values.

I'll leave it to you to discover how the input driver file is constructed, suffice it to say that it is vaguely similar to the Font file but has a GEOS file type of 10.

Icon Control

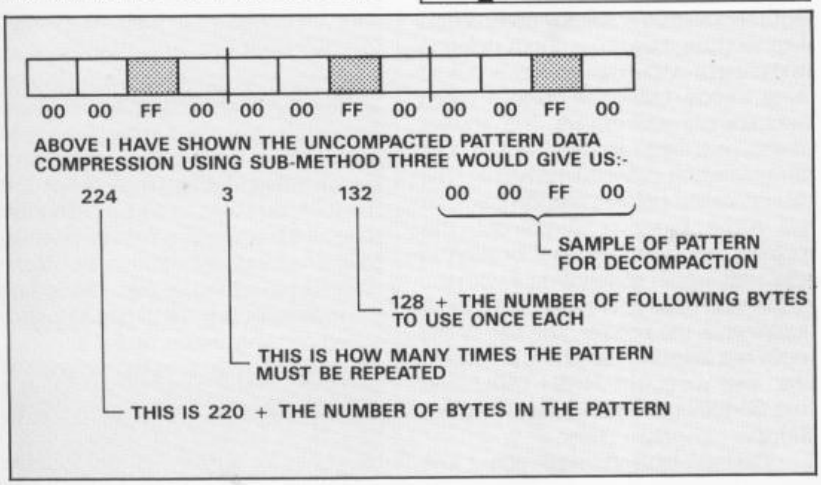
Icons and menu's are one of the most important things to understand in GEOS and provide a very powerful user interface. GEOS uses tables of data called Icon tables to set up an icon for display on the screen, GEOS only uses one routine for controlling icons. These icon tables tell GEOS where to put the icon, how many icons to display, the size of the icons and where to go if the icon is clicked. GEOS is told where the required icon table is by loading the start address into the word at $02. Then a call to the DoIcons routine in the GEOS kernal at $C15A will initialize and display the icons in the table. Clicking on one icon may cause a new table to be accessed in the same way and display a new set of icons.

The table is laid out as follows, the first byte of the table tells GEOS how many icons are to be displayed using this table. The next three bytes tell GEOS where to place the mouse after the icons have been displayed, X position followed by Y position. This is useful if you want to place the mouse over the default icon. Then the individual icon data follows with icon one's entry beginning with the pointer for the start of the Icon graphics data for icon one in word form. Now comes the X position of the icon in bytes (0-39) followed by the Y position in pixels, these refer to the top left corner of the icon. The width in bytes and the height in pixels are next, but are stored on the same byte, e.g. 1 byte wide and 12 pixels high = 112. Finally we must tell GEOS the pointer for the routine which must be called if the icon is clicked, this routine is called the service routine. The data for icon two, if there is one, follows. The service routine can do anything it wants without restriction and simply ends with an rts. When an icon is clicked GEOS will make two pieces of information available to a service routine. In byte $02 it will store the number of the selected icon and in $03 it will store the value $00 for a single click and $FF for a double click. The point of these bytes, if you are wondering, is to allow two icons to direct control two one service routine if their requirements are similar, and for the service routine to know which of the icons was selected. Also the double click principle means that each icon can have two meanings as used in Geopaint. There is one final global variable used with icons and it is at location $84B5. Bit 7 of this variable if set tells GEOS to flash the icon when it is selected, like the undo in Geopaint. If bit 6 and not bit seven is set then the icon is inverted by GEOS when selected like on the desktop files. If neither bit is set then GEOS will not give any visual indication of selection, but if required the service routine could give an audible indication.

Finally, but by no means the easiest bit for me to explain, we take a look at the icon graphics data and how it is stored. If you have never come across this method of bit map storage before, then I hope my explanation is understandable. All bit-mapping graphics in GEOS are compacted in a way designed to save memory space and also reduce disk access time for loading such data. An icon is bit mapped graphic and is therefore compacted. GEOS compacts data in the count bit-map pair format using three different methods, I shall call them sub-methods from now on. In the first sub-method which is identified by the count data byte being between 0 and 127, the count byte tells GEOS how many times to repeat the following bit-map byte before continuing. The second sub-method, which is identified by the count byte being of a value between 128 and 220, tells GEOS to use the following amount of bytes (quantified by the count byte) as bit-map bytes. The final sub-method uses a count byte between 221 and 255 and tells GEOS to use the following pattern data of N bytes, where N. ????? The amount of times to repeat the pattern is in the following byte and the pattern following that in one of the first two sub-methods.

To try a second explanation I have drawn it graphically in figure 1.

All this data is also scanned horizontally across the graphic starting at the top left hand corner instead of the cards the C64 memory uses, a way in which I find screen data much less of a headache. Of course if you want to produce your own icons and don't want to know how they are stored you can use the icon editor and make life easy. Finally a word of warning, as if you develop an application for GEOS which does not use icons you must still include the table and data for one icon as GEOS always assumes you will use at least one. Next issue we will probably look towards the first menu structure and I will be providing the last 4 fonts.



Hashing It

The definition of 'HASH' in the Concise Oxford Dictionary has got nothing whatsoever to do with computers, so you may well ask why this article is appearing in a computer, and not a cookery, magazine. Well the answer is this. To hash something means, basically, to make it smaller. Well that is the method that is used in hash tables. First of all I will explain exactly what a hash table is.

A hash table is simply a table, holding the location where the data is to be stored, calculated by using an equation and performing this equation on the data, the result of this equation is called the HASH value. Effectively the equation is making the data into a smaller piece of information and using this piece to locate the original piece in the array. For example:

Let's pretend we have an array which has 10 spaces. We want to store numbers which are in the range 1000-2000, the has equation could well be:

HASHVAL=INT((data/100)-10)

where data is the value we want to store. This equation, for all legal entries i.e 1000-2000, gives a result which is:

0=>result<=10

the result is between, and including, 0 and 10.

Inserting Data Into A Hash Table

Now we have established what our equation is, let's try to enter a few values into the array. We shall enter the following: 1900, 1000, 1600, 1500.

If you work out the HASH values of the above you will find that they are 9,0,6 and 5. So the number 1900 will be placed in location 9, 1000 in location 0, 1600 in location 6 and 1500 in location 5. (See Diagram 1.0).

The computer would take hardly any time to calculate the equation, so data can be inserted into an easily retrievable location almost instantaneously. So when you come to search for your data, the same equation is performed and the data is found, if it exists, almost immediately. This all sounds very good, but, unfortunately, a few problems have to be overcome before the hash table can be used efficiently. I will demonstrate just one of those problems now. What would happen if I wanted to enter the number 1502 into the table above? The equation would return the hash value 5, but this location is already occupied so I cannot use it again, without deleting the current data. The answer is not so easy to see, but quite obvious when you know (what isn't?). The answer is this, you have another field in your array, making it two dimensional. The field is called pointer and contains, as it's name suggests, a pointer, this pointer points to the next location in the array which contains data with the same hash value as the data in the same location as the pointer. The size of the array has to be increased and separated into two parts, the HASH TABLE and the OVERFLOW TABLE. (see Diagram 1.1)

DIAGRAM 1.0

ADDRESS	CONTENTS
0	1000
1	EMPTY
2	EMPTY
3	EMPTY
4	EMPTY
5	1500
6	1600
7	EMPTY
8	EMPTY
9	1900
10	EMPTY

The hash table should be about 10% of the entire table, hash+overflow. Now when we try, once again, to enter 1502 into the table we can. The computer first hashes 1502 and gets the result, 5, it then looks at location 5, it isn't empty so it looks at the pointer, which is 0. The computer sets this pointer to the next free location, the first location of the overflow table, which is 11. Then the computer stores the value 1502 at location 11. (See Diagram 1.2). The next free location then becomes 12, whenever another collision occurs, location 12 will be used.

If we were to add another piece of data which returned the hash value 5, then the computer would first look at location 5, finding this to be occupied, it would look at the pointer of location 5, the pointer contains 11 so the computer would go to location 11 and look there, finding it is occupied it would look at the pointer. The pointer at location 11 is null, contains zero, so the computer would make this pointer equal to the next free location, 12, and store the data in location 12.

If an efficient hash equation is used, the one I have used is not at all efficient, then the number of collisions, on average, is less than 2, so the computer, whilst searching, would only have to look at a maximum of two records before finding the required one.

Sorting A Hash Table

Before we go any further I want to tell you one thing which you cannot do with hash tables. You cannot sort them, if you were to sort a hash table then all the pointers would be pointing to the wrong data and none of the data would be retrievable. This is a bit inconvenient, but the amazing search speed certainly makes up for it.

Searching For Data In A Hash Table

To search for an item in a hash table

If you have ever wondered how you can get 4 crocodiles into a mini then this article on Hashing is for you

By Steve Burgess

you must choose which field, if there is more than one, is to be the key field, this field is the one which is hashed all the time, it is hashed when you are entering data, searching for data & deleting data, this field is the field by which records can be accessed. Once you have established which is the key field you can enter data. Then you can search for an item.

Let's say we have a massive database containing the names & telephone numbers of people we know. The key field would be the NAME. If I wanted to find the phone number of Fred Bloggs, I would enter Fred Bloggs, this would be hashed, found and his number almost immediately displayed.

The programming behind this is a little more complicated. It is correct that the key field would be NAME, and the name of the person needs to be entered. The entered value is then hashed, in the case of strings the hash

DIAGRAM 1.1

ADDRESS	CONTENTS	POINTER
0		
10		
11		
100		

HASH TABLE

OVERFLOW TABLE

equation would have to use the ascii values of the characters or the length of the string, and the computer looks at the location indicated by the result of the equation, if FRED BLOGGS is stored here, then the telephone number is displayed. Otherwise the computer looks at the pointer field, if it is zero then the name FRED BLOGGS is not in the database, if it is a value other than zero then the computer looks at the location indicated by this value, if FRED BLOGGS is stored here they hey presto, we've found our man. This continues until the data is found or the pointer is zero. When the pointer is zero it means that there is no more data with the hash value of the search string, which means FRED BLOGGS is not in our database.

Deleting Records From A Hash Table

Another slight disadvantage with hash tables is the difficulty one encounters when trying to delete items from them. There are a number of different methods, which include re-arranging the pointers to accomodate the deletion, or having an array to store the free location addresses. While these methods may be ultra efficient, I prefer to fill the deleted record with a null value but to keep all the pointers intact. When data is entered which has the same hash value as the previously deleted data, then the space made available by the deletion is used.

Imagine we wanted to delete 1500 from our table (see diagram 1.2), the computer hashes 1500 and comes up with 5, it then deletes the data but doesn't touch the pointer. That location is now free to have more data stored in it. Let's add 1522. The computer hashes and gets 5 as the result, it finds that location 5 is empty to it puts our new data, 1522, in that location. Therefore you can read the data, as if nothing had been changed, without the computer crashing.

DIAGRAM 1.2

ADDRESS	CONTENTS	POINTER
0	1000	NULL
1	EMPTY	NULL
2	EMPTY	NULL
3	EMPTY	NULL
4	EMPTY	NULL
5	1500	11
6	1600	NULL
7	EMPTY	NULL
8	EMPTY	NULL
9	1900	NULL
10	EMPTY	NULL
11	1502	NULL
12	EMPTY	NULL
13	EMPTY	NULL
14	EMPTY	NULL
15	EMPTY	NULL
16	EMPTY	NULL

Using The Program

Load the program by typing Load "Capitals",8 or use the menu. After a short delay you will be presented with a menu:

1..ADD RECORDS
2..SEARCH FOR RECORD
3..DELETE RECORD
4..END PROGRAM

The reason there is a delay just after running the program is that the computer is reading in the data, which is a list of countries and their capitals.

You can add your own countries, if you want to, by selecting option 1.

If you want to search for a capital of a country you select option 2, you will be prompted to enter the country name and the capital will then be displayed.

If you want to delete an item from the array select option 3. You will be prompted to enter the country name and then asked if you are sure. The record will then be deleted, but if you rerun the program it will reappear because the data is stored in DATA statements.

To end select option 4.

Well that's it, the possibilities for hash tables are endless. I hope you can find a way to use them which will be of some benefit to you.

Multi-Sprite

An easy to use Sprite Utility that allows up to 24 sprites to be programmed independently of each other

By Jason Finch

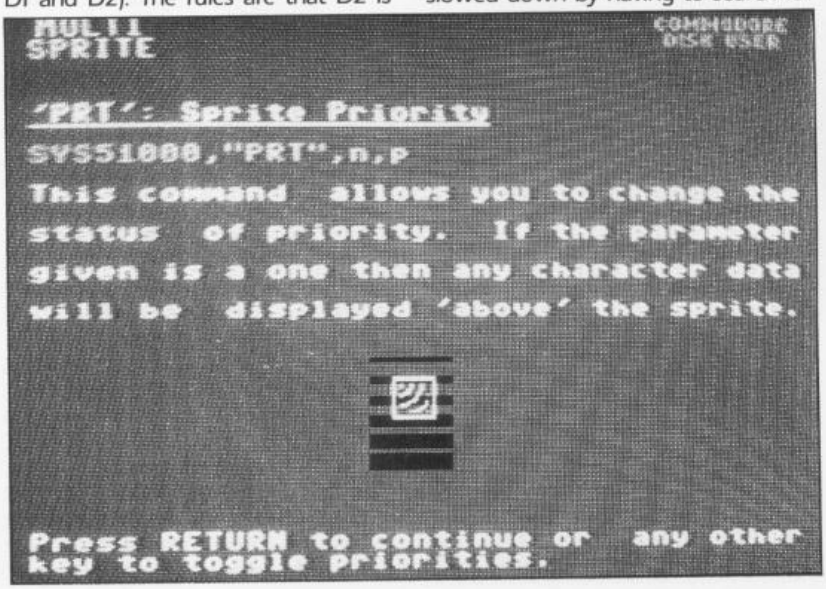
With regard to the graphics facilities of the Commodore 64, sprites are undoubtedly the most versatile. For most applications the eight sprites provided as standard are adequate. However, there are various other programming projects in which this poses a rather annoying restriction. If you are not familiar with the workings of "raster-scan" graphics then you will face a problem, one that is increased considerably if you have a limited understanding of machine language.

It is for this reason that I have programmed "Multi-sprite", the most versatile multiple sprites system commercially available for the Commodore 64. Many readers will have seen one or more programs that fall in the category of "infamous 64 sprite extravaganzas". The sort of thing I am talking about is the display of 64 men, mindlessly running across the screen. These programs introduce the concept of more than eight sprites but, like the "Commodore 64 Programmers Reference Guide", they provide no details of how to create more than eight individually controllable sprites.

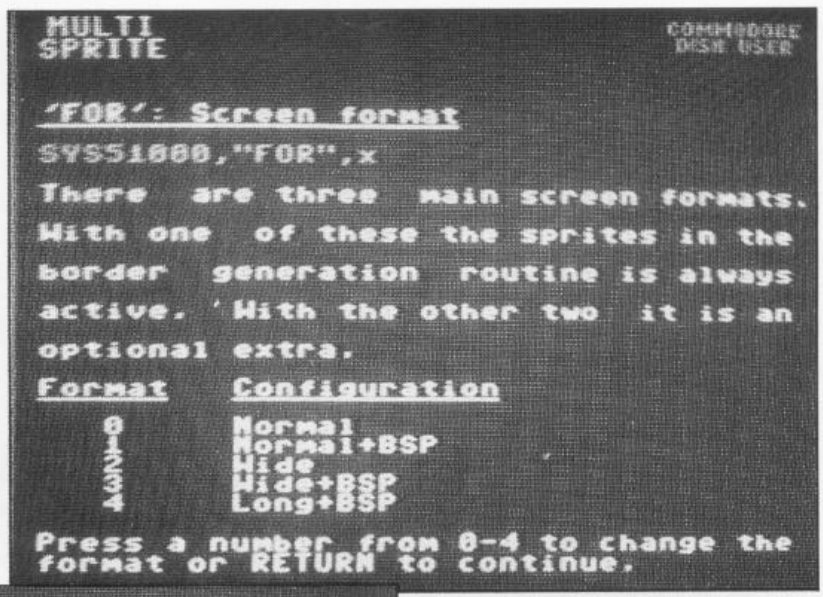
The program on the disk will allow up to twenty-four sprites to be positioned, coloured, expanded and so on completely independently of each other. Even the priority that a sprite has over character data can be differed for each of the twenty-four sprites. As an optional extra you can have sprites in the upper and lower borders. When using more than eight sprites the program also allows you to cure the vertical wraparound of sprites that you may have experienced in the past using this technique. You can also choose from a range of border and background configurations.

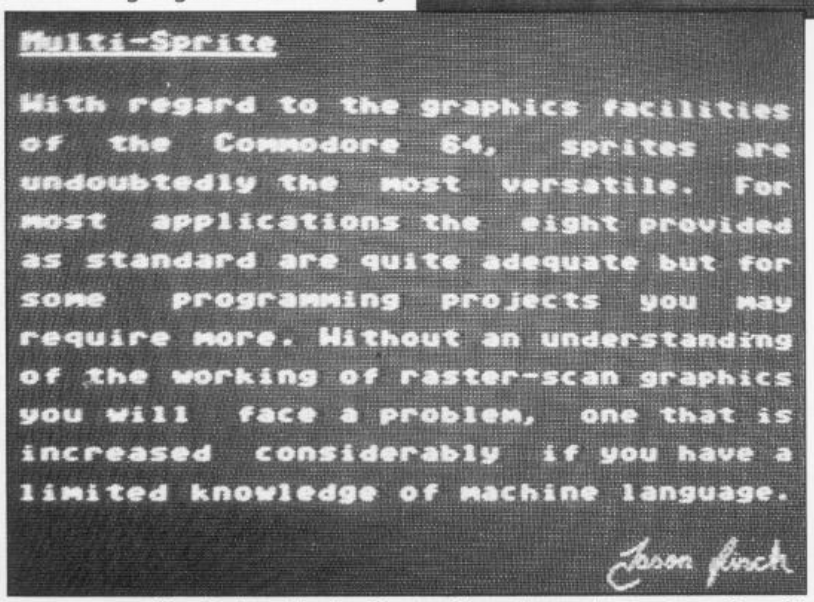
Although each sprite is able to be controlled individually, you must remember that there are, essentially, up to three blocks of eight sprites. When using 16 sprites there is one division vertically and thus two blocks, and with 24 sprites there are two divisions and three blocks. These divisions are not visible but suffice it to say that any one sprite from a particular block of eight should not be moved vertically beyond the boundaries specified by the position of the vertical splits. There is, however, an exception which is documented later on. There are only two general rules that should be adhered to when 24 sprites, and thus two divisions, are being used (I shall call these divisions D1 and D2). The rules are that D2 is an offset from D1 and has a minimum value of 32 pixels, and that D1 + D2 must equal no more than 149. You should also remember that when D1, the first division, has a value of zero, it will lie about three lines into the main screen.

Each individual characteristic of a sprite is changed by a set of simple to learn three letter commands. To prevent the computer's interpreter being slowed down by having to search for



Extended BASIC recognition characters (such as the 'at' sign or exclamation mark) and to leave as much memory free as possible, I have used a simple method for carrying out the instructions to control the sprites. You simply type: SYS51000, "<commands>" ,x,y – where 'x' and 'y' represent parameters needed by the program. If this method becomes tedious, I may provide an update that would allow COMMAND, xy, y to be entered directly without the need for a SYS call. This will only happen though if I have a great number of requests (send any to me via my problem page). I have also structured the program in such a way that each command is as easily accessed from your own machine code programs as it is from BASIC. The commands and their meanings together with necessary parameters are listed below. For details of how to call each from machine code you should consult the program 'M-S TECHNICAL' on the disk. This program will also inform you of the locations that you should "PEEK" to read the position, etc of any of the sprites.





'INT' initiates the necessary interrupts to provide you with up to 24 sprites. It takes no parameters. Alongside its main function the command will reset all positions, colours, pointers and so on for all 24 sprites to a default value of zero. The characteristics of the standard eight sprites are then copied into the routine's stores. The screen format will also be reset to standard configuration and no extra sprites will be provided. When the routines are in operation the normal POKE commands associated with both sprite manipulation and screen colour changes cannot be used.

'OFF' is the opposite to 'INT', switching off all interrupts. It also takes no parameters and should be used any Input/Output devices are accessed. This will restore the standard screen configuration of a border around the entire screen and will limit the number of available sprites to the usual eight until the 'INT' command is used again.

"FOR" sets the screen configuration with regard to border and background. It takes one parameter which must lie between 0 and 4. Any higher value will result in an illegal quantity error being generated. A value of zero sets the standard configuration. A one will allow sprites in the border, 2 provides a 'wide' screen, 3 does the same but with sprites in the border, and 4 creates a 'long' screen effect, again with sprites in the border. All the routines allowing border sprites cater only for sprites in the upper and lower border regions. Then using border sprites, ensure that location 16383 always contains a zero (POKE 6383,0). This is due to the bug in the Video Interface Chip that enables such effects as sprites in the border.

"VDU" simply allows you to change the colour of the border, background and ink. It therefore takes three parameters, each of which must lie between 0 and 15. They are the border colour, background colour, and the cursor colour, respectively.

"POS" is used to alter the physical position of any of the 24 sprites. It takes three parameters which, in order, are the sprite number (0-23), the horizontal position of the sprite (0-511), and the vertical position (0-255).

"COL" changes the colour of a specified sprite. It takes two parameters. These being the sprite number (0-23) and the colour (0-15) respectively.

"S12" allows you to change the horizontal and vertical expansion of a sprite. The first parameter is the sprite number and the second is a value from 0 to 3 which indicates the expansion required. 0 means no expansion, 1 is horizontal only, 2 is vertical only, and

3 is both combined.

"ENA" enables (or disables) a sprite. It takes two parameters – the first being the sprite number and the second being either a zero or a one. A zero switches the sprite off and a one switches it on.

"PRT" changes the priority that a sprite has over character data. Its two parameters take the same range as for "ENA" but here, a zero indicates that sprite data has priority and a one allows any character data to have priority.

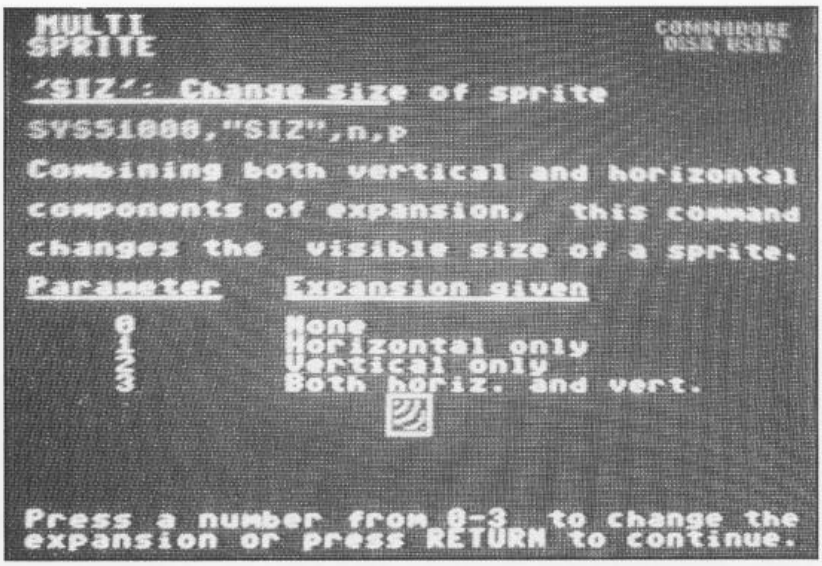
"MUL" is another on/off command, this time associated with multicolour sprites. It takes the same range parameters as for the "ENA" command. A zero as the second parameter switches multicolour mode off for that particular sprite and a one switches it on.

"CL1" sets multicolour register one to a specific colour for one of the three blocks of eight sprites. It takes two parameters – the first between 0 and 2 inclusive indicates the block number and the second indicates the colour and must lie between 0 and 15.

"CL2" is the same as "CL1" but changes the second multicolour register for the defined block.

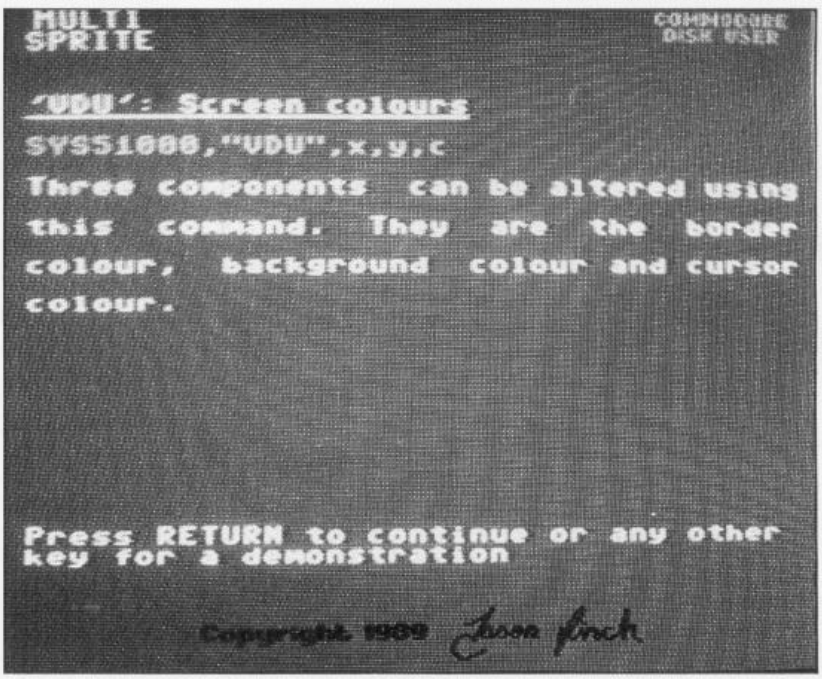
"POI" again takes two parameters and is used to change the sprite block pointer. In other words the block of 64 bytes that the computer should read to get the information for the sprite definition. The first parameter is the sprite number and the second is the block number.

"BKP" allows you to instruct the computer to change the base address from which the sprite pointers are read. Usually this is 2040. This command, directed towards the more advanced programmer, takes one parameter that falls between 0 and 65535. Most addresses will never usually be used for the base address of the sprite pointers but there are a few. As I hope you would appreciate, whenever changing the sprite block pointers the computer uses the value stored by this command (a default of 2040 is set when the "INT" command is used) and so if you have set an invalid number the computer may crash. For example, if you have set the base address to zero then if you were to instruct the computer to change the pointer of sprite zero, the value would actually be written to address 0 in the computer's memory – this would almost certainly cause the computer to lock



up! The value given should actually be one less than the desired value. So, to change back to the default base address you would give a value of 2039.

"ADD" takes one parameter and enables you to add a further eight sprites up to a total of 24. It takes one parameter which indicates the position of the vertical split that has to be created to allow more sprites. If adding to produce sixteen, the parameter assumes the value for the first and only split and has a range of 0 to 149. If adding eight sprites to produce a total of 24 then it assumes the offset from the first division for the second division. If the first division is 117 or more (ie: so that if positioned with a minimum offset the second division will be out of the range outlined earlier) then an error will be created. If this is so, then the first division will have to be moved using the "DIV" command described below. It is important that you remember that values associated with commands like this do NOT represent the actual position of the split. One further note on the positioning of divisions, with any of the three formats



allowing border sprites, the maximum range should be reduced from 149 to no more than about 130. If you do increase it then a lot of flickering will occur which cannot be rectified by the timing command below.

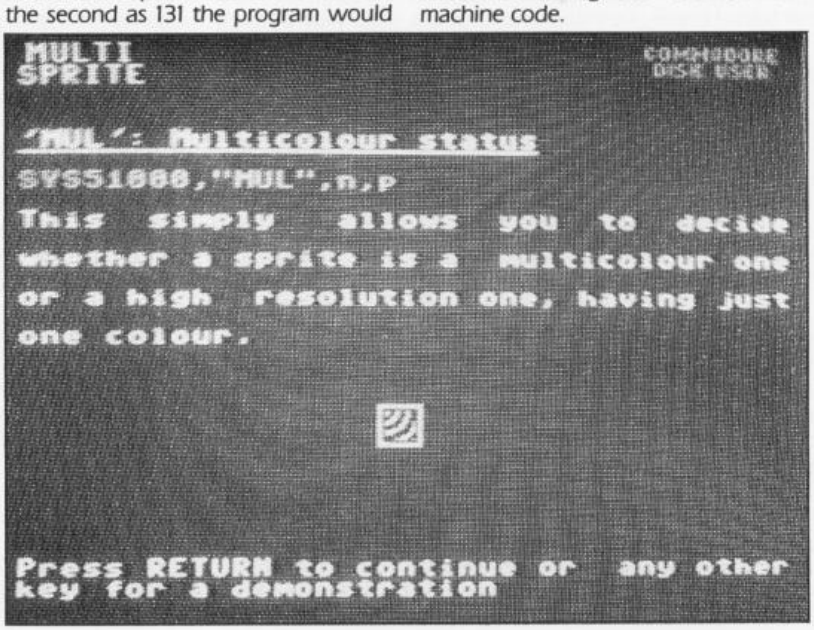
"REM" is the opposite of "ADD", removing eight sprites down to a minimum of the usual eight. It needs no parameters.

"DIY" is the command used to alter the vertical position of existing splits. If using 8 sprites then this routine is irrelevant and will generate an error when called. When using sixteen the routine takes one parameter, the position of the only division that will be present. When 24 sprites are used, the routine takes an extra parameter. The first being the position of the first division and the second being the offset for the second. Both forms of the command require parameters that fall within the correct ranges that I described earlier.

"TIM" is a specialised command that is associated with timing loops that are necessary to avoid small flickers that occur at the position where border colour changes to background colour and vice versa. Usually these flickers will not occur unless sprites fall near the aforementioned regions. The values will generally need to be found by trial-and-error. The routine needs two parameters, each in the range 1 to 20, although you should never really need a value of more than about ten. The first is the timing loop at the top of the screen and the second is the timing loop at the bottom of the screen. For reference, the default values are 4 and 1 respectively.

"COM" takes two parameters and is a very powerful command enabling 'common' sprites. It effectively creates a double of one sprite from one block and places it on one or more other blocks. Every time the sprite from the lowest block number (zero is the uppermost on the screen) is moved, coloured, etc, then so is its double in the other blocks(s). The first parameter is the basic sprite number. There are eight possible sprites in each block and hence this value must be between 0 and 7. The second parameter is the block selector and has a range of 1 to 3. A one will take a sprite from block zero and create a double in block one. A two will take a sprite from block one and create a double in block two, and a three will merge the latter two, taking a sprite from block zero and duplicating it in both blocks one and two. Using the command with a second parameter of one and then again with a parameter of two (assuming the same basic sprite number) can be used to create the same effect. With any arrangement, only one of the sprites is visible at any one time allowing, for example, a game using twenty-four sprites to have six sprites in the upper section, six in the middle section with two (the player and his weapon) common to both of the latter blocks, with a further eight in the lower section that could be used for scoring, etc. Usually as one sprite is moved over the split into the next block it disappears or takes on the form of the corresponding sprite in the next block . Using this technique (without employing common sprites) it is possible to produce high resolution sprites that have two colours vertically. This can be used in titling but is not practical for moving objects. Now back to the purpose of this command because you may not want a colour split or something similar to occur. With the "COM" command the sprite will appear to move continuously down the screen with no flickers (the BASIC demo has an example of the command in action). By adding 128 to the parameters to produce a range of 129-131 inclusive, you can instruct the computer to toggle common sprite status. For example with the first parameter as a three and the second as 131 the program would allow sprites 3, 11 and 19 to be controlled separately again if they had previously been defined as common. Repeating the command with the same parameters would then reverse this action. You should note that this command should be used before the sprite characteristics (expansion, colour, etc.) are set up.

That about covers all the commands and the details that you need to operate the program. Just now a short summary of the command names: INT, OFF, FOR, VDU, POS, COL, SIZ, ENA, PRT, MUL, CL2, POI, BKP, ADD, REM, DIV, TIM. On the disk you will find file "M-S SAVE/REL" you can save the code to your own disk and, despite the code being rather long (49152-51287 incl), you can change the start address to which it loads. Not quite as simple as it sounds, for the computer will also amend the code as it goes so that it will execute correctly at your chosen address. For this reason the saving of code unfortunately takes a considerable while longer than the simple save. If you have a machine code monitor and do not wish to relocate the code (and you have an impatient streak!) then I suggest you use that to save the code. The program, "H-S TECHNICAL", is for other machine code programmers who may wish to use the routines from within their own programs. Upon RUNning, it will list to screen or printer all you need to know for calling the routines from machine code.



Directories Explained

Gain a better understanding into the workings of the directory and it's related tracks

By Jason Finch

The disk directory is possibly the most important part of a disk, telling the drive, and indeed the user, what sort of file a program is, how long it is and, more importantly, the program name and where to find it on the disk. The directory, despite storing all this data for up to 144 files, remains an extremely manipulative feature of a disk. Once you have mastered the art of reading individual sectors from the disk – and writing them back without losing the whole lot – this manipulation becomes increasingly easier.

First though I shall explain some "disk basics" to ensure that I am not confusing you any more than is necessary! Each disk is divided into 35 concentric rings, or tracks, with each track containing from 17 to 21 sectors – track 1 is the outermost and contains 21 whilst track 35 is the innermost, containing only 17. For a complete list of the number of sectors of each track consult your disk drive's manual. Each sector contains 256 bytes of data. With the correct knowledge the user can examine and/or alter any byte from any sector of any track on the disk. The directory is located on track 18, beginning at sector zero. Although each disk theoretically contains 683 separate sectors the directory track is not available for general usage and so we are left with 664 sectors – or blocks – free.

The BAM

No, it has absolutely nothing to do with those words that light up your television screen when the Joker gets kicked in the face by Batman. It is merely the abbreviation for Block Allocation Map – the most important part of the directory track, and indeed the disk – more important than the files themselves. "Don't be stupid" I hear you cry – but it's true. Without the BAM, each time you saved one file you would probably overwrite another. The BAM keeps track (sorry!) of the status of each of the 683 sectors on the disk – recording which are free and which are used. Without it the drive wouldn't have a clue as to which sectors could be written to. I have never seen the BAM data explained fully so after some investigation I can provide you with as detailed an explanation as you will ever need – or want!

It is found on track 18, sector 0 and it occupies 140 bytes (4 for each of the 35 tracks on the disk) beginning at the fifth byte. While we're here I should mention that immediately following the BAM is the header and disk ID information. But back to the BAM. Each set of four bytes is made up as follows: the first byte is a "checksum" and it keeps record of not which, but how many sectors are free on the track to which it refers. The next three bytes inform the drive of which particular sectors are free. The rest requires a bit of detail and is easier to understand if seen graphically, and so I have written a program to do just that. Details are provided later in this article of how to go about loading it. Suffice it to say that you will be able to read, and alter, the BAM yourself if necessary without too much hassle. Just remember, though, that the BAM is updated every time a file is saved or the disk is validated.

Byte by Byte

The first two bytes of any sector represent pointers to the next track and sector to be read. If the track pointer byte (ie: the first byte of the sector) is a zero then the end of the file (or directory) has been reached. If this is so then the sector byte informs the Disk Operating System of how many bytes in that sector are actually used. Thus if you were to examine the first two bytes of track 18, sector 0 you would find that they were 18 and 1 respectively, telling the drive to read track 18, sector 1 next. This is the only instance on a disk where sectors are read consecutively — elsewhere on the directory sectors are chained in steps of three and on all other tracks in steps of ten. This system is common to all sectors but here we shall concentrate specifically on those on track 18 – the directory.

Sector zero is special, storing the BAM and the disk header and identification code information. We shall deal with these later. Let us deal now with sectors containing the information concerning individual files. Each block of 256 bytes in the directory consists of eight entries, each of 32 bytes. These represent the following:

Bytes 0-1:	not used (unless first two bytes of block)
2:	file type ($81=S equential, $82=Program $83=User, $84=Relative)
3-4:	track and sector where file is located you could use these to examine a file from within another program.
5-20:	filename (ASCII codes)
21-29:	not used
30-31:	blocks used by file in LO/HI format. for files under 256 blocks use byte 30

You can access these bytes by using a disk editor, but what happens if you

want to access them from within a program. For example you may want a word processor to check whether or not a certain file is sequential. So why not do it yourself with the aid of the drive's direct access commands? All necessary details are provided below and in the accompanying progress.

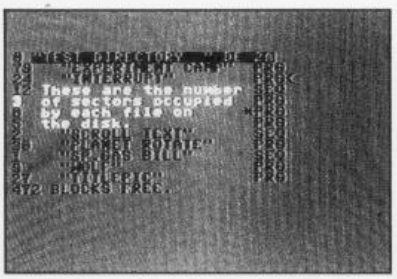
First you must always open the drive's command channel and one to read and write to a buffer (in that order); all communications between you and the drive will be via this buffer.

```
OPEN 15,8,15: OPEN 8,8,8,"*"
```

Then use the 'U1' command to read a specific block into the buffer. Note that if you use the B-R command then data will be corrupted as the first byte of the block is used as a byte count:

```
PRINT#15, "U1 8 0 18 1"
```

In the above command, the '8' represents the buffer channel, the '0' the drive number, the '18' is the track – the



directory – and '1' is the sector. That sector is then read into an internal buffer. You should then use the 'B-P' (Buffer Pointer) command to access whichever byte(s) you want:

```
PRINT#15, "B-P: 8"; B
```

where 'B' represents the byte number (0-255). So now you have seen how to open the necessary channels, read a sector of the directory containing information about up to eight files, store it in the buffer and to instruct the buffer pointer to look at one byte of your choice from that sector. To read this byte use the following:

```
GET#8,A$: IF A$="" then
A$=CHR$(0)
```

The latter statement is necessary in case the byte is a null ($00). A length of characters can be read using a simple FOR...NEXT loop. This is the method for reading the filename. When reading the size, type and location of a file use *N=A$(A$)* and remember that the check for a $00 byte is still necessary.

When you wish to change any information within the buffer, simply point at the correct byte using the Buffer Pointer command and then:

```
PRINT#,A$;
```

If you have altered the entire buffer, point at byte zero before writing information back. The semicolon (;) in the above example is vital – should you forget it a carriage return will be written. This can be especially annoying if you have changed a file type because the carriage return byte ($0D) will overwrite the start location of the file. Once you have written back the information to the buffer, either as single bytes or as a string of up to 255 characters, it is essential that the first two bytes are the track and sector pointers to the next sector. It is only necessary to rewrite them if they have been changed, of course:

```
PRINT#8, CHR$(TRACK);
CHR$(SECTOR);
```

You can then transfer the information from the buffer to the disk using the 'U2' command, the opposite of 'U1'. (The B-W command has the same problem as B-R) The channels should then be closed, buffer first:

```
PRINT#15, "U2 8 0 18 1": CLOSE 8:
CLOSE 15
```

The 'U2' command assumes the same parameters as 'U1' and could equally have been written as:

```
PRINT#15, "U2"; 8; 0; 18; 1
```

Now that you know the exact procedure we shall perform some experiments! Make sure you have an old disk handy – jut on the off-chance that you make a mistake. Experimenting on valuable disks, even when you are sure you know what you are doing, can be very distressing if you press the wrong key and realise too late – I speak, unfortunately, from experience!

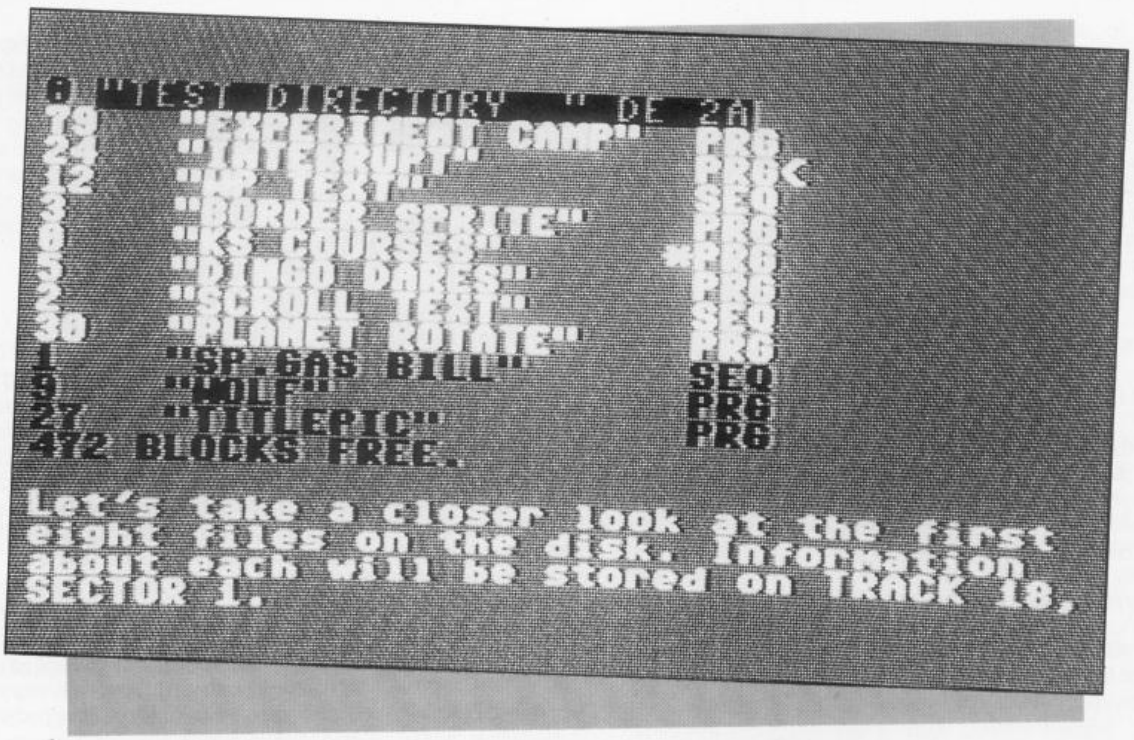
On the disk

Load and run the program filed as "DIR EXPLAINED". I would advise you then to remove the CDU disk – although I have faith in my programming, accidents do happen – such as that freak power cut in the middle of a save and replace when your entire week's programming efforts disappear in a cloud of expletives!

There are two menus, from menu one you can load in three other programs – the main demonstration, the explanation of the workings of the BAM (something, as I have mentioned, that I have never seen done before), and a directory sort. The latter sorts the directory alphabetically according to the ASCII codes. I would strongly suggest never doing this to a CDU disk as all the dividing entries accumulate at the beginning. It is mainly for this reason that the program enables you to see what the finished article will look like before it is committed to disk.

Option four is a detailed directory list, not only giving the usual details but also the start location of each file. Option five is called Program Tracer. The first sector occupied by the first file on the disk is read. The pointers to the next are provided and then this is read, and so the process continues. Option six swaps the filenames only of the first two files on the disk. Do it again to get back to normal. All the aforementioned options contain detailed step-by-step instructions of how to carry out each one from within your own programs. Option seven calls up menu two.

Menu two is more of a DOS+ menu with features such as unscratch, protect a file, change header and ID and also a powerful feature I have never come across in Disk Toolkits before – obliterate file. It sounds rather ominous but it simply searches for your chosen filename and then, if found, you are asked to confirm that you wish to destory the file. The computer then erases the entry from the directory and



completely clears every sector on the disk used by that file – use with extreme care.

Option one of menu two is a software write-protect of individual files. Commodore never reveals more than they have to and it's a shame there is no BASIC command like this. Supply a filename and let the computer take over. Full instructions are again provided. Options two and three allow you to change the disk header and identification code respectively, with the latter being up to five characters. Option four is an unscratch command.

If you realise your mistake in time use this. When a file is scratched the file type is simply changed to a zero – that's it. Change it back to whatever it should be (usually $82 for a program file, 130 in denary) and hey presto! This, however, will not work with files that have been wiped completely using the obliterate a file option – option number five. Option six allows you to enter the standard disk commands such as rename, initialise, validate, scratch and so on. One thing to note – where you would normally use a colon, use a semicolon. This is because the prompt is a simple BASIC input command and anything after colons is ignored. Option seven simply recalls menu one.

I shall repeat one more time that ALL of the options from both of the menus have extremely comprehensive step-by-step instructions. When you are asked to press any key, the SHIFT/LOCK button can be depressed to prevent the computer pausing.

This is extremely useful during an 'obliteration' where a long file can take a few minutes to be completely destroyed.

You should note, in particular, the way in which the next track and sector pointers are read in and used in the reading of the next sector. To recap, you can locate any individual criterion of a file, such as type, size, etc., using the following syntax after having located the appropriate sector:

```
PRINT#15, "B-P: 8"; FNHx32+B
```

Each program is a standard BASIC program and can be loaded by typing LOAD"(filename)", 8 and then typing RUN when the READY. prompt reappears. There are four separate programs, each rather long due to the detailed instructions, and these are filed as follows:

"DIR EXPLAINED" is the main menu. You can load in the other programs or perform disk manipulating operations.

"DE1" is the main demonstration about directories,

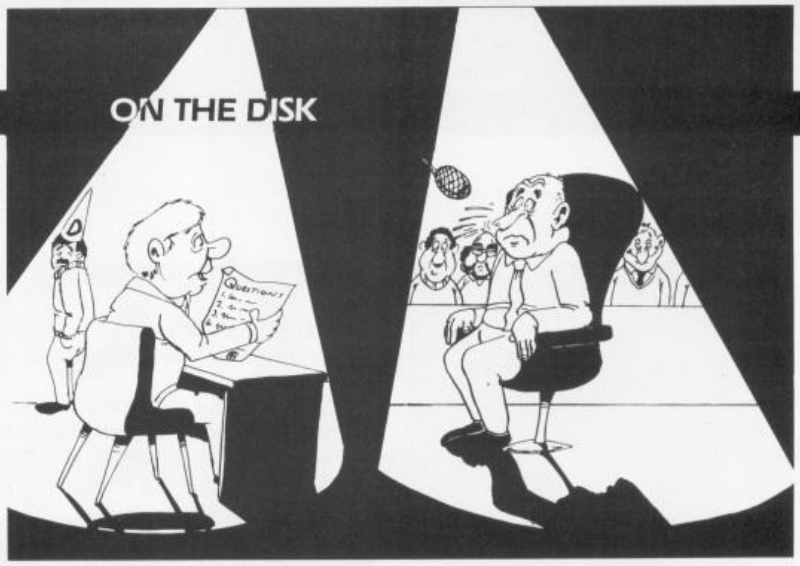
"DE2" is the alphabetical directory sort, and

"DE3" is the explanation of the Block allocation Map.

Each of the latter three can be loaded from within the main menu program.

Concluding

Once you have examined all the facilities provided by each of the programs, you should be able to confidently change program file types, etc (there must be some reason), track down (sorry!, again) a program's location on the disk, and even find out where in the memory it loads to (yes – it is covered in the main demo), rename a file without the simplicity and boredom of a normal RENAME command, change the program order on a disk... and still be able to load the directory and programs on it without creating error messages that Commodore had no idea existed – well, one can live in hope!



If you know how many golf balls are on the moon or which English King had a tattoo then you might well be an expert on trivia – so have a go at **Trivia Challenge,** you might learn something absolutely useless!

Trivia Challenge is a game in which there are over two thousand questions, mostly on quite useless facts. Answering questions correctly earns points which, at the end of a game will be converted into a code, which along with the score may be used to enter the **Commodore Disk User Trivia Challenge** competition.

Trivia Challenge – Part 2

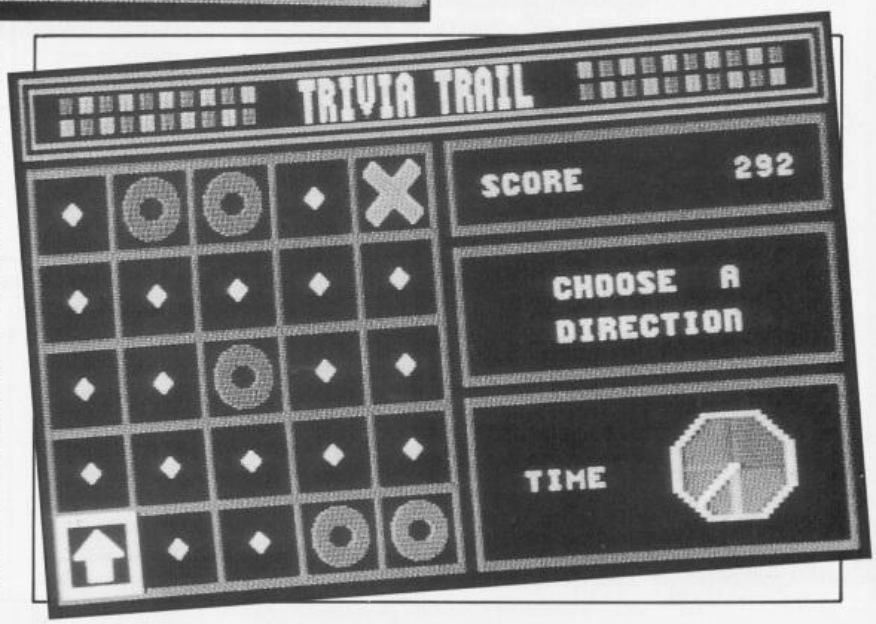
There are three rounds in the game, which repeat and get more difficult as the game progresses.

The first of the three rounds is called *tic-tac-trivia* – a simple(!) game of noughts and crosses, in which the player lays down crosses against the computer, playing as noughts. The players alternatively select a square and the player is asked a question. A correct answer puts a cross in the square, a wrong answer puts a nought. Three crosses in a row and the player wins, three noughts and the computer wins and the game ends. In the event of no winning (or losing) lines then the numbers of each symbol are totalled and the greater one wins.

The clock on the tic-tac screen shows the time available to choose a square – the SPACE BAR is used to move a cursor to the required square and the RETURN key will play that square.

Success at noughts and crosses leads to round two which is called *trivia trail* and is played on a 5x5 grid. The player starts in the lower left hand corner of the grid and must move to the upper right hand corner, which is marked with a cross. Each move is earned by correctly answering a question, after which an arrow will appear and can be changed by pressing the SPACE BAR to select the required direction, then the RETURN key to make the move.

At the start of this round, each square on the board contains either a diamond marker, each of which are worth ten points when landed on, or a bonus marker (a nought) which



Here is the second part of the Trivia Challenge game.
By Keith Suddick

carries extra points, or free moves or a clock reset when landed on for a second time.

The clock on the trivia trail screen shows the time available for the entire game. There is a time penalty for any wrong answers. The players can only win by getting to the "cross" and loses only by time out.

The last of the three rounds is called *trivia flash* and is simply a rapid sequence of questions – 2 for each time through the three rounds. There is no time limit but ALL the questions have to be answered correctly or the game will end. After "trivia flash" the game returns to "tic-tac-trivia" and so on but it will be slightly more difficult each time around.

When the questions are displayed, four possible answers are always shown, only one is correct. The keys 1 to 4 OR the four functions keys may be used to select an answer. A clock is also shown at the bottom of the screen and starts with 25 time units shown, a correct answer will earn 4 points for each time unit remaining when the key is pressed – time out will end the game.

As already mentioned, when the game ends a competition code will be shown along with the final score, BOTH the score and code are required to enter the competition, so if you wish to enter, copy them down carefully – they will only be shown on the screen for a limited time so have a pen and paper ready.

As should become obvious, the disk needs to remain in the drive throughout the game, so it only remains to say Good Luck!

About the Competition

CDU in conjunction with the author, challenge you all to enter to better my score.

Once we have published all 3 parts of Trivia Challenge and you have linked them together to form the game (see Getting It In), you will have a chance to pick up a colour monitor for your C64. At the end of this article you will see a competition score sheet. Simply fill out the details and send your entry to:-

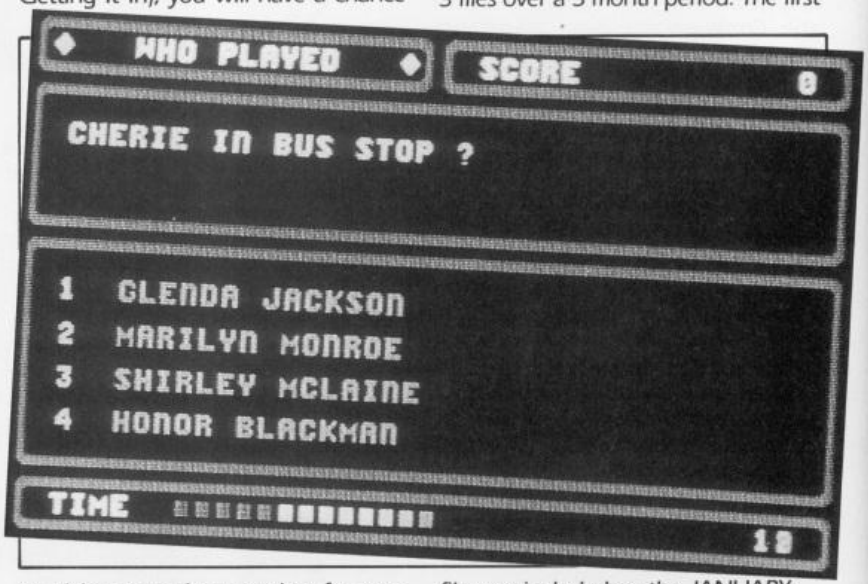
Trivia Challenge,
CDU,
Argus House,
Boundary Way,
Hemel Hempstead,
HP2 7ST.

Closing date for the competition will be **30th June 1990**. (This should give our Australian readers a fair crack of the whip!!!). Only one entry per person will be allowed. Employees of Argus Specialist Publications and their relatives are exempt from the competition.

Getting It In

Trivia Challenge will be presented as 3 files over a 3 month period. The first file was included on the JANUARY disk of CDU. On this month's disk you will find the file named Trivia 2 and the Trivia Install program. Once you have all 3 Trivia files, load and run Trivia Install and have a blank formatted disk ready. Simply follow the onscreen instructions to make a final version of Trivia Challenge.

Note: Due to the way the program is saved out in the final version, you will not be able to load and see the directory. In order to run the final version simply put the disk in the drive and type **LOAD "TRIVIA", 8,1**



COMMODORE DISK USER TRIVIA CHALLENGE

NAME _____

ADDRESS _____

_____

_____

_____

SCORE

COMPETITION CODE

PLEASE USE BLOCK CAPITALS

Tomorrow's Tomorrow

Is this the rantings of a mad man, or is it logically possible. Read on and make up your own mind. (How many more readers are like this – Ed???)

By John 'Bones' Simpson

You're not going to believe this! But I swear it's all the truth, only the colours have been changed to prevent investigation.

I was ambling home, across the field at the back of the estate.. you know, just behind the corner shop. The stars were shining pretty bright, not a cloud in the sky, and the gossamer swathe of the Milky-way glowing in the way that only the Milky-way can. The moon was somewhere else, and I could just make out the wing-lights of the southbound shuttle.

I never had much on mind... which is pretty much the norm, when all of a sudden I heard a kinda synthaesized whistling sound. I knew it wasn't Mally on his Yamaha, 'cause he was away in Holland. Anyway it was out in the middle of the field! Next moment I saw shaft of laser-look-alike light belch out of what seemed to be a star, and zap into the ground not even twenty paces away. The ground sort of shimmered for a mo, then everything was back to 'normal'.

I cautiously went over to the 'spot' to investigate and even before I arrived I could make out this sort of greeny luminous glow emanating from within the grassy tufts, and faintly, almost inside my head, I could hear the Police! No, not the law, but that ancestral music machine with a sting at the head, I couldn't quite make out what song I was hearing, but as I got closer to the ersatz green-glow, the tune began to slip slowly into place. As I reached the spot, there was this translucent olive-green object, almost buried in the grass. The music in my head increased in volume as I pushed the object clear of the grass with the toe of my right trainer. It was about the size and shape of a small cola-bottle. Then the music became clear.

Hesitantly I picked it up. As I did so the music abruptly stopped. It was warm to the touch and I had the impression of vast distances... but not of space. I studied the 'bottle' more closely in the starlight. It was made of some smooth, softish, yet unyielding, material, and as I watched, it changed colour to a flat black. I turned it over, looked at the end, which definitely resembled a bottle-cap, tossed it up in the air, giving it a flick so that it spun end to end, caught it in mid-spin and studied the other end. Yes, it was a sort of petrified bottle that changed colour, camed out of the sky accompanied with light and music, and was soft yet unyielding to the touch, and made me feel good! I rammed it in my pocket and sprinted for home.

My mum was just making a spot of tea when I got in, breathless and excited by my find.

"Cuppa tea, son?" she grunted.

"Na," I replied, "Gotta project to do."

So saying, I rushed off upstairs to my room.

I closed the door, pulled out the little bottle and stood it on the desk, in front of 'Old faithful', my 64, pulled up a chair, sank into it and stared at my find, pondering. I hadn't pondered for too long, when all of a sudden the bottle emitted what can only be decribed as a self-satisfied purr, then immediately began to melt. Well, I'm not sure if melt is the right sort of description... but that's what it looked like, yet controlled. It sort of started forming a pool, but a pool with straight edges and right-angle corners. Quite soon it was wafer thin, and about the same size as the 64.

I leaned forward to touch it with my index finger, but it sort of 'yelped', and without loosing its rectangular, straight-edged form, somehow withdrew itself from my probing finger. With a squeal of what can only be a described as delight, it flowed with incredible speed straight up and over the key-pads of 'Old faithful', then, like water draining through sand, it filtered through the molecular structure of the 64 and disappeared within!

Stillness and silence, a fly just missing my gaping mouth. My mind draining numb with the speed of the events. After a while the muffled sounds of 'Coronation Street' slowly filtered through the floorboards from the downstairs living-room, and

twenty, maybe thirty seconds later, the little red on/off indicator light at the top right of the 64 blipped on.

"Huh!" The power was still switched off.

After a long moment, and with me staring like a dodo at the red-light, it started to flash on and off. Thrice rapid, thrice slow, thrice rapid, thrice slow. After another long moment of watching, it dawned on me. The international S.O.S. signal! I could not believe it – here I was in my bedroom with a bottle from 'up-there', which played mental music, changed shape, and filtered into my computer, which, without power, was now flashing the S.O.S. signal at met. What to do? What did it want?

Suddenly it became quite clear, turn on the monitor, Drongo!

So I did. And a fanfare of music errupted through the speaker which could easily have been recorded by Pink Floyd at Wembley Stadium. We all know that SID, as a three-voice synth, is good, but we didn't know he was this 'vicious'! Then the screen started shimmering, racing through multicolours, soft and pastel, steel and enhanced. The bog-standard sixteen colours of 'Old faithful' errupting suddenly into mega-mills, and which, after a few moments, rapidly disolved into the head and shoulders of the most beautiful being – human, yet not quite, and not of Glory knows what sex. The resolution was far better than a 35mm photo. Had some of the 'bottle' flowed from the output of the 64 to the input of the monitor?

The music subsided into the background and the head on the monitor stared at me, its perfect lips moving.

"Hi, Bones", a golden-coloured voice murmured gently from the monitor's speaker, honey-like.

An insane laughter belched through my lips, like bubbles from out of the hot mud of a geyser.

Let's not talk about Max Headroom! The resolution was so brill that it seemed as though there was a real head within an empty monitor-box.

"Have I shocked you?" asked the golden-voice, a whimsical look appearing on the face in the monitor.

"H-huh, huh", I stuttered.

Over the next few days, Jahvkal, the little green bottle, ran a series of mind-boggling programs through the C64, which was now in some form of symbiosis with the friendly Jahvkal, and answered my many questions, and told me of the great daze to come.

Basically, it seems that Jahvkal, the messenger, has arrived here from a future time to prepare us for the coming changes which, apparently, start tomorrow.

Over the next decade it seems there will be some astounding and profound breakthroughs in both computer hardware technology, and software engineering.

Apparently the two-gate logic system will shortly become redundant and will be replaced by three-gate logic. Trinary! Whereas now 1 = an on switch, or 'closed-gate', and 0 = an off switch or 'open-gate', tomorrow it's going to include a -1 which will equal a gate that is neither 'open' or 'closed'. It gets a teeny bit baffling here, but as Jahvkal demonstrated, and with the most sublime graphic displays, the 'new' trinary system emulates the neuron-mass biological processor much more closely. Further to this – each trinary gate is in itself a mini-microprocessor.

I questioned the mathematics of all this. For example in binary things are fairly simple with progression in powers of two (ie, 2, 4, 8, etc). In trinary the progressions are in powers of three (3, 9, 27, etc). But as Jahvkal demonstrated, the whole system is computer designed, and with operating systems which carry such massive and advanced interpreters that us humans don't need to know, or worry about such matters. English, or any other language, in any dialect will be readily understood.

The age of the silicon chip is out! The upper limits of chip technology being simply too poor to contain enough gates, and far too slow to operate quickly enough, Mega-mills of instructions requiring multi-parallel execution in nano-secs. The new systems will operate within a sub-nuclear level using substances which, in this day and age, do not yet exist.

It seems that a new theory put forward by an unknown, unscientific and part-time programmer in the latter decade of the 20th Century, which at first was completely ignored and labelled 'crank', but then siezed upon by the scientific community when certain principles became understood, gave birth to the new mechanics of Quark manipulation. Apparently this theory suggested that quarks are the base unit of all matter, and themselves being nothing other than complete infinite universes existing within other different dimensions, only their effect apparent within our universe, (and, as it happens, our universe itself being a mere quark within another!).

Although earlier researchers and theorists suggested many different kinds of quarks, there are, in fact, only three types. These are the Up Quark (UQ), the Down Quark (DQ), and the Steady State Quark (SQ). The first two belonging to the expanding/contracting universe type, and the latter the static, unchanging type. The energy output of the UQ is said to be positive, the DQ negative and the SQ neutral. Although, it seems, negative, positive and neutral are misnomers in the case of quark mechanics.

Because the quark doesn't actually occupy any space within our universe new substances will be created which can contain an astronomical number of quarks. In fact the entire 'substance' will contain only quarks and what will be called 'drones', which, incidentally are constructed from the quarks themselves. There was, at first, a colossal weight problem but this was eventually overcome as the time dimensions were mapped more thoroughly!

Within the 'substance', known more affectionately as 'Qsub', the quarks will be arranged into three-dimensional matrices. Viewed graphically they will resemble with an uncanny familiarity, those first computers, namely the Chinese Abacus, albeit, racks upon racks of them numbering in megabits.

Sub-proton engines are to be designed, tested and constructed using the new quark technology (Qtech) and will loosely be called droned. They draw their energy requirements directly from the i/o of the UQ/DQ. A full-sized, fully operational drone occupying much less space than a clumsy electron. These new engines emitting, with the use of radical new interrupt request techniques short-length energy waves – not laser light, this being too gross. The new term 'Quark Energy, Source Temporal Emittance Repulse Systems', 'qesters' being employed. The drone emits a gesters for a milinan, but then will require two milinans to re-energise. So, the system employs a triad of drones working in unison to main-

tain steady 'qesters'. The end result is the shifting of a quark within a matrix array. This is similar, but not the same as, opening and closing gates.

The flow of the ultimate, and strongest of all the physical forces, that which binds together quarks, now being easily and simply manipulated. Altogether it resembles a billiard cue (the questers) striking billiard balls (the quarks) into pockets (positions in the matrix).

With, as next to damn it, unlimited volumes of Qmats (Quark matrices) available, vast storage systems become possible within 'pin-head' sizes giving rise to enormous operating systems, vast executable programs, and virtually unlimited mega-memory data storage buffers. The whole system being self-operating, self-sufficient, and self-reparable.

Once the new 7th generation computers become operational, they will rapidly design and construct mobile units within which to contain themselves, and designing new 'tough' materials so to do. A new age of robotics begins. Work for humans will rapidly become redundant, as will poverty, warfare, strife, and tears.

And then, in the early decades of the twenty-first century, the new dawn arrives. Communication with the non-living! It seems, from the lessons Jahvkal gave to me, that the true life force which exists within all living structures, and gives living things their 'soul', or 'spirit', their element of self-identify and awareness, is an energy source from without the physical confines of matter.

Apparently this is where we come from before we are born, and where we go to when we die. Not just as humans but all forms of life. Trees, ants, zebras and aardvarks. The whole unified element of life-force organised the whole manipulation of biological structure from the very beginnings of earth-time, learning from trial and error and building up a vast base of knowledge. At first LifeForce (LF) could muster enough energy to manipulate simple molecular compounds to create the earliest of life forms (amoeba and suchlike), then, by infiltrating the central processing units (brains and simple nervous systems), LF was able to manipulate the minute amounts of chemicals and electrical forces needed to run self-structuring and modifying programs. The story of evolution.

Eventually LF had produced 'us', humans. Here physical evolution stopped and mental evolution took over until eventually evolution could be speeded up by negating the need to evolve more complex biological structures.

Humans would 'invent' things! These 'things' could be structured using non-biological materials, and so gain a rapid increase into the gathering of knowledge, in preparation to the BIG quest. To journey out to the stars!

The early 7th generation computers will rapidly put together a working model of a device for communication with LF on the 'other side'. Small personal models will then be mass-produced, which will strangely resemble Sony Walkmans, and these will be given to each and every human. Anti-birth drugs will be created and automatically seeded into the atmosphere to ensre the eventual ceasure of life on the planet. But at the same time millions of 'new beings' of all types and descriptions will be manufactured, and constructed from the new materials which will now be synthesized, and grown in vats! Beings who will be able to go anywhere, even to the centre of a sun, and who will not require the input of normal solar energy sources. All energy requirements being drawn from the new Qtech. Beings who will not only feel and savour all of life's great beauties, but who will be all these things. Nothing will be lost, not a sunset, not even a dodo. All will be the great group mind of LF. As each new being emerges from the production vats, LF will flow into it and give it self-identity, and life. All beings in sympathetic harmony and communication with each other, and with LF.

All the knowledge gained by LF contained in one massive group mind, and yet with each participant a complete independent individual, able to draw upon this vast reservoir of knowledge and experience, and all united in a common unity and quest. At last, after billions of years the planet will become one, and will be a ready to step forward in a new direction of exploration, to boldly go where no one has boldly gone before.

At the time of Jahvkal's visit as messenger, LF was in the process of constructing vast city-sized and living starships prepared and ready for the voyages outwards into the unknown. But first one of the most important tasks was at hand and Jahvkal had been the volunteer to do the job. To defy all the known and understood laws of pyhsical structure. To be swept back into the past in order to give the world the information it needed to start the new Qtech on its way.

And that is why I have written this article. I know you'll find it incredibly hard to believe and full of flaws, but that's because I'm just an average joe, non-scientific, uniknown, and a part-time programmer. Jahvkal's time will soon be up, and then it will be automatically drawn back along the time channels to its own present.

Me, I'm in the process of writing this new program for my humble 64 which should eventually outline the whole basis of Qtech.

Might take me a while though!

COMSOFT (Coldstream)Ltd. Comsoft House. Coldstream. Scotland. TD12 4BS.

ACCESS Telephone No.0890 3232 Fax No.0890 2927 Carriage: Orders less than £10.00 please add £1.00 VISA

AMSTRAD PC1512 COMPUTERS

	R.R.P	Our Price
SD MONO MONITOR	458.85	412.97
SD COLOUR MONITOR	573.85	516.47
DD MONO MONITOR	573.85	516.47
DD COLOUR MONITOR	688.85	619.97
20mb HARD DRIVE/CARD FITTED		375.00
32mb HARD DRIVE/CARD FITTED		375.00

AMSTRAD PC1640 COMPUTERS

	R.R.P	Our Price
SD COLOUR MONITOR	588.85	529.97
SD ENHANCED COLOUR MONITOR	803.85	723.47
SD MONO MONITOR	573.85	516.47
DD COLOUR MONITOR	803.85	723.47
DD ENHANCED COLOUR MONITOR	918.85	826.97
DD MONO MONITOR	688.85	619.97
HD COLOUR MONITOR	1207.50	1086.47
HD ENHANCED COLOUR MONITOR	1148.85	1033.97
HD MONO MONITOR	918.85	826.97

AMSTRAD PC1000 COMPUTERS

	R.R.P	Our Price
PC1256 SINGLE DRIVE	882.50	776.25
PC1512 SINGLE DRIVE	1437.50	1293.75

AMSTRAD PC2000 COMPUTERS

	R.R.P	Our Price
PC2086 20mb HARD DRIVE	882.50	776.25
PC2086 DOUBLE DRIVE	682.50	569.25
PC2086 SINGLE DRIVE	537.50	485.75
PC2286 40mb HARD DRIVE	1207.50	1086.75
PC2286 DUAL DRIVE	997.50	879.75
PC2386 40mb HARD DRIVE	2127.50	1914.75

AMSTRAD VGA MONITORS

	R.R.P	Our Price
12" HIGH RES COLOUR MONITOR	882.50	776.25
12" MONO MONITOR	171.35	154.22
14" COLOUR MONITOR	284.35	257.72
14" HIGH RES COLOUR MONITOR	518.35	464.72

PORTABLE COMPUTERS

	R.R.P	Our Price
PPC512 DOUBLE DRIVE	631.35	568.22
PPC512 SINGLE DRIVE	458.85	412.97
PPC640 DOUBLE DRIVE	746.35	671.72
PPC640 SINGLE DRIVE	573.85	516.47

AMSTRAD WORDPROCESSOR

	R.R.P	Our Price
AMSTRAD PCW8256	401.35	361.22
AMSTRAD PCW8512	536.35	484.72
AMSTRAD PCW9512	573.85	516.47

HOME COMPUTERS

	R.R.P	Our Price
AMSTRAD CPC 464 COLOUR	299.99	269.99
AMSTRAD CPC 464 MONO	199.99	179.99
AMSTRAD CPC 6128 COLOUR	399.99	359.99
AMSTRAD CPC 6128 MONO	299.99	269.99
ATARI ST 520FM	299.99	269.99
COMMODORE 64	149.99	134.99
COMMODORE AMIGA	399.99	359.99
SPECTRUM +2	149.99	134.99
SPECTRUM +3	199.99	179.99

PRINTERS

	R.R.P	Our Price
AMSTRAD DMP2160	159.00	151.05
AMSTRAD DMP3250/I	194.35	184.63
AMSTRAD DMP4000/I5" WIDE	286.35	272.03
AMSTRAD LQ3500 24 PIN	286.35	272.03
AMSTRAD LQ5000 24 PIN	458.85	435.91
STAR LC10	228.85	217.41
STAR LC10 C64 Interface	328.85	217.41
STAR LC10 COLOUR	297.85	282.96
STAR LC10 COLOUR C64 Interface	297.85	282.96
STAR LC24-10 24 PIN	343.85	326.66

PRICES FOR OTHER MAKES PLEASE PHONE

JOYSTICKS

	R.R.P	OUR PRICE
CRYSTAL CLEAR	14.99	13.46
CRYSTAL TURBO CLEAR	16.99	15.29
JY2 JOYSTICK	14.95	13.46
QUICKSHOT 2	9.95	8.96
SUPER PROFESSIONAL	16.99	15.29
THE ELITE	14.99	13.49

BLANK MEDIA

	R.R.P	OUR PRICE
3" CF2 DISCS (10)	29.95	26.96
3.5" DOUBLE DENSITY (10)	24.95	22.46
3.5" DS/HD DENSITY (5)	22.87	20.58
3.5" SINGLE DENSITY (10)	22.95	20.66
5.25" COLOURED DISCS(10)	17.95	16.16
5.25" D-SIDE D-DENSITY By 3M (10)	26.00	23.32
5.25 S-SIDE/D-SIDE By Dikblog (10)	9.95	8.96
5.25 S-SIDE/D-SIDE By Dikblog (20)	19.95	17.96
5.25" HIGH DENSITY (10)	17.61	15.87
C15 BLANK CASSETTES (10)	6.90	6.21
3/3.5" SPARE DISC BOXES (10)		4.00

BLANK MEDIA STORAGE

	OUR PRICE
3.5" DISC STORAGE By Touchmaster Holds 80	12.95
CASSETTE STORAGE RACK Holds 20 CASSETTES	1.50
DD100L HOLDS 140 DISCS	14.95
DD100L HOLDS 100 DISCS	12.95

COMPUTER DUST COVERS

	OUR PRICE
AMSTRAD CPC464 COLOUR MONITOR	10.76
AMSTRAD CPC464 MONO MONITOR	10.76
AMSTRAD CPC6128 COLOUR MONITOR	10.76
AMSTRAD CPC6128 MONO MONITOR	10.76
AMSTRAD PCW8512 COLOUR DISPLAY	12.59
AMSTRAD PCW8512 MONO DISPLAY	12.59
AMSTRAD PCW8256/8512	13.49
AMSTRAD PCW9512	13.49
AMIGA A500	6.99
ATARI ST FM	6.99
COMMODORE 64	4.99
SPECTRUM +2	4.95
SPECTRUM +3	5.85

PRICES FOR OTHERS MAKES PLEASE PHONE

PRINTER DUST COVERS

	OUR PRICE
AMSTRADDMP2000/2160/3000/3160/3250 PRINTER	5.38
AMSTRAD DMP4000 PRINTER	4.95
AMSTRAD LQ3500 PRINTER	4.95

PRICES FOR OTHER MAKES PLEASE PHONE

AMIGA SOFTWARE

	R.R.P	OUR PRICE
ACTION AMIGA	29.99	25.49
ALTERED BEAST	24.99	21.24
BANBOK KNIGHTS	24.99	21.24
BATTLE VALLEY	24.99	21.24
BEAST	24.95	29.74
BRIDGE PLAYER 2150	29.95	25.45
CAPTAIN BLOOD	9.99	8.49
CHESSMASTER 2000	19.99	16.99
CHESSMASTER 2000	24.99	21.24
DAILY D.HORSE RACING	19.99	16.99
DEEP SPACE	9.99	8.49
DELUXE MUSIC CONSTRUCTION SET	69.99	59.49
DELUXE PAINT 2	69.99	59.49
DELUXE PAINT	24.99	21.24
DYNAMITE DUKE	24.99	21.24
EARL WEAVER BASEBALL	24.99	21.24
EMPEROR OF MINES	24.99	21.24
FAST LANE	19.99	16.99
FRIDDISH FREDDY	24.99	21.24
GAMES SUMMER EDITION	24.99	21.24
INSTANT MUSIC	24.99	21.24
IRON TRACKERS	19.99	16.99
LANCASTER	19.99	16.99
MARBLE MADNESS	24.99	21.24
MEGA PACK 2	24.99	21.24
MR HELI	24.99	21.24
OIL IMPERIUM	24.99	21.24
OOZE	19.99	16.99
PAPERBOY	19.99	16.99
PASSING SHOT	24.99	21.24
PREDATOR	29.95	25.45
QUEST FOR TIME BIRDS	24.99	16.99
RALLY CROSS	29.99	25.49
RED LIGHTNING	24.99	21.24
ROBOCOP	19.99	16.99
SAINT AND GREAVSIE	19.99	16.99
SCARFHOST	19.99	16.99
SHINOBI	24.99	21.24
STAR WARS TRILOGY	14.99	12.96
SKYFOX	19.99	16.99
STRIP POKER 2	24.99	21.24
STRIKER	24.99	21.24
SWORDS OF TWILIGHT	24.99	21.24
TARGHAN	24.99	21.24
TRIAD 2	14.99	12.96
THE BARDS TALE	24.99	21.24
THE SEVEREN CITIES OF GOLD	24.99	21.24
WAYNE GRETZKY ICE HOCKEY		
XENON 2		

AND MANY MORE TITLES PLEASE PHONE

AMIGA EDUCATIONAL

	R.R.P	OUR PRICE
AB ZOO	14.99	12.96
DISCOVER CHEMISTRY	19.99	16.99
DISCOVER MATHS	19.99	16.99
DISCOVER NUMBERS	19.99	16.99
DISCOVER THE ALPHABET	19.99	16.99
PUN SCHOOL 2 Age 6-8	19.99	16.99
PUN SCHOOL 2 Age 8+	19.99	16.99
PUN SCHOOL 2 Age Under 6	19.99	16.99

AMIGA BUSINESS/UTILITIES

	R.R.P	OUR PRICE
ADIUM	39.95	33.96
AMAX	134.95	114.71
AMAX With ROM	269.95	232.46
AMIGA UPGRADE	89.99	76.50
ANIMAGIC	79.95	67.96
AUDIO DIGITISER (MONO)	37.99	23.79
AUDIOMASTER 2	79.95	67.96
BIG BAND	169.95	144.46
C-64 EMULATOR 2	49.95	42.46
C-LIGHT	49.95	42.46
DAY BY DAY	29.95	24.90
DEVPAC 2	59.95	50.96
DISCMAC	39.95	33.96
DRAW 2000	224.95	191.21
E-TYPE	39.95	33.96
GPA BASIC VERSION 3	64.95	55.21
HANDY SCANNER (24)	679.95	577.96
HANDY SCANNER (TYPE 2)	299.99	254.99
HI SOFT BASIC	79.99	67.99
HOME ACCOUNTS	29.95	25.46
HUSH PRINTER	119.95	101.96
ICON PAINT	19.99	16.99
K-DATA	49.95	42.46
K-ROGET	29.95	25.46
K-SEKA	49.95	42.46
K-SPREAD	59.95	50.96
KINDWORDS V.2	49.95	42.46
MAILSHOT	24.95	21.21
MAILSHOT PLUS	49.95	42.46
MIDI MASTER 500	38.49	32.72
MODELER 3D	79.95	67.96
PAGESETTER	113.85	96.77
PERSONAL ACCOUNTS PLUS	34.45	29.29
PERSONAL TAX PLANNER	39.95	33.96
PHOTON PAINT	69.99	59.49
PROTEXT	99.95	84.96
PUBLISHERS CHOICE	99.95	84.96
SECOND DRIVE	134.95	97.71
SMALL BUSIN ACC.(CASH)	80.45	68.38
SMALL BUSIN ACC.(XTRA)	134.94	97.69
SONIX	64.95	55.21
STUDIO 24	155.00	131.75
TRACK 24	75.00	63.75
X-CAD DESIGNER	113.85	96.77
ZOETROPE	99.95	84.96

COMMODORE 64 ENTERTAINMENT

		R.R.P	OUR PRICE
AIRBORNE RANGER	CASS	14.95	12.70
AMERICAS CUP CHALLENGE	CASS	9.99	8.49
APOLLO 18	CASS	9.99	8.49
AZIMUTH HEAD ALLIGNMET KIT	CASS	8.99	7.65
BALLBLAZER	DISC	14.95	12.70
BATMAN THE MOVIE	CASS	9.99	8.49
BATMAN THE MOVIE	DISC	14.99	12.74
BATTLE CHESS	DISC	14.99	12.74
BEYOND DARK CASTLE	DISC	19.99	16.99
BLAZER	DISC	14.95	12.70
BATTLE COMMANDER	CASS	9.99	8.49
BEACH HEAD 2	CASS	9.99	8.49
BIG NAME BONANZA	CASS	9.99	8.49
BOULDERDASH CONSTRUCTION KIT	CASS	9.99	8.49
CALIFORNIA GAMES	CASS	9.99	8.49
CARD SHARKS	DISC	14.95	12.70
CHESSMASTER 2000	DISC	14.99	12.74
CYRUS 11	DISC	13.95	11.85
CHERNOBYL	CASS	9.99	8.49
CHESSMASTER 2000	CASS	9.99	8.49
CHOLO	CASS	14.99	12.74
CRICKET MASTER	CASS	8.95	7.65
CYBORG	CASS	9.99	8.49
DANTE'S INFERNO	CASS	9.95	8.46
DAVE	CASS	7.99	6.79
DEEP STRIKE	CASS	7.99	6.79
DEMON STALKERS	CASS	9.95	8.46
DRAGON STALKERS	DISC	14.95	12.70
DISCOVERY	DISC	14.95	12.70
DRAGON SKULLE	CASS	8.99	7.65
DYNAMITE DIX	CASS	9.99	8.49
DYNAMITE DIX	DISC	14.99	12.74
DISK 50	DISC	14.99	12.74
EARTH ORBIT STATIONS	DISC	16.95	14.40
EMPIRE	DISC	14.95	12.70
EROLON	CASS	7.99	6.79
EXPLORER	CASS	9.99	8.49
FIRETRAP	CASS	9.99	8.49
FOOTBALL MANAGER 2	CASS	9.95	8.46
FOUR GREAT GAMES 2	CASS	3.99	3.39
FRANKENSTEIN	CASS	9.99	8.49
FIELD OF FIRE	DISC	14.95	12.70
FRANKENSTEIN	DISC	14.95	12.70
GAMES CRAZY	CASS	12.99	11.04
GAMES CRAZY	DISC	14.99	12.74
GRAHAM GOOCH ALL STAR CRICKET	DISC	12.95	11.07
GUNSHIP	CASS	14.95	12.74
GUNSHIP	DISC	19.95	16.95
GHOSTCHASER	CASS	9.99	8.49
G GOOCH ALL STAR CRICKET	CASS	9.95	8.46
GREYZOR	CASS	8.95	7.60
HACKER II THE DOOMSDAY PAPERS	DISC	14.99	12.74
HEAD OVER HEELS	DISC	14.95	12.70
HIGHROCK	CASS	8.95	7.60
I ALIEN	CASS	9.99	8.49
IMPLOSION	CASS	8.95	7.60
INSTANT MUSIC	CASS	9.95	8.46
INSTANT MUSIC	DISC	14.95	12.70
JAPAN SCENERY	DISC	14.95	12.70
JETBOYS	CASS	9.99	8.49
LIFEFORCE	CASS	9.99	8.49
LORDS OF CONQUEST	CASS	9.95	8.46
KAYLETH	DISC	14.95	12.70
LORDS OF CONQUEST	DISC	24.95	21.20
MANDROID	DISC	14.95	12.70
MARBLE MADNESS	CASS	9.99	8.49
MARBLE MADNESS	DISC	12.95	11.07
MASTERS OF THE UNIVERSE	CASS	9.99	8.49
METROCROSS	DISC	14.95	12.70
MR HELI	CASS	9.99	8.49
MR HELI	DISC	12.99	11.04
MIND CONTROL	CASS	8.95	7.60
MINI PUTT ACCOLADE	CASS	9.99	8.49
NEMESIS THE WARLOCK	CASS	8.95	7.60
NEMESIS THE WARLOCK	DISC	12.95	11.01
NINJA HAMSTER	CASS	9.99	8.49
OIL IMPERIUM	DISC	14.99	12.74
PASSING SHOT	CASS	9.99	8.49
PASSING SHOT	DISC	12.99	11.04
PASTFINDER	CASS	9.99	8.49
PHM PEGASUS	CASS	8.95	7.60
PLASMATRON	CASS	9.99	8.49
PLASMATRON	DISC	14.95	12.70
PIRATES OF THE BARBARY COAST	DISC	14.95	12.70
POWER AT SEA ACCOLADE	DISC	14.95	12.70
POLAR PIERRE	CASS	9.99	8.49
RAINBOW WARRIOR	CASS	9.99	8.49
RAINBOW WARRIOR	DISC	14.99	12.74
ROGUE TROOPER	DISC	14.95	12.70
RAMARAMA	CASS	8.95	7.60
SAINT AND GREAVSIE	CASS	9.99	8.49
SCENERY/DISK 1	DISC	12.95	11.01
SCENERY DISK 11	DISC	14.95	12.70
SCENERY/DISK 3	DISC	12.95	11.01
SCENERIE/DISK 4	DISC	12.95	11.01
SCENERIE/DISK 5	DISC	12.95	11.01
SCENERIE/DISK 6	DISC	12.95	11.01
SHINOGE	CASS	9.99	8.49
SHINOGE	DISC	14.99	12.74
SHOGUN	CASS	9.95	8.46
SHOOT EM UPS	CASS	9.99	8.49
SIGMA 7	CASS	9.95	8.46
SKATE OR DIE	CASS	9.99	8.49
SKY RUNNER	CASS	9.95	8.46
SPORTING TRIANGLES	CASS	9.99	8.49
SPORTING TRIANGLES	DISC	14.99	12.74
STAR WARS TRILOGY	CASS	12.99	11.04
STORY SO FAR Vol.4	CASS	12.99	11.04
STORY SO FAR Vol.4	DISC	14.99	12.74
STUNT CAR	CASE	9.99	8.49
STUNT CAR	DISC	14.99	12.74
STRATTON	CASS	9.99	8.49
STAR WARS TRILOGY	DISC	19.99	16.99
STRIKE FLEET	DISC	14.95	12.70
SUPER DRAGON SLAYER	CASS	4.99	4.24
SKY RUNNER	DISC	14.95	12.70
SKYFOX II	DISC	14.95	12.70
TAG TEAM WRESTLING + KARATE	CASS	9.99	8.49
TARZAN	CASS	9.95	8.46
TERRY'S BIG ADVENTURE	CASS	9.99	8.49
TERRY'S BIG ADVENTURE	DISC	9.99	8.49
THANATOS	CASS	7.95	6.75
THE BARDS TALE II	DISC	16.95	14.40
THE CONSULTANT	DISC	24.95	21.20
THE HITS	DISC	17.99	15.29
THE HITS	CASS	9.99	8.49
THE SECRET DIARY OF ADRIAN MOLE	CASS	8.95	7.60
THE TRAIN	CASS	9.95	8.46
THEY SOLD A MILLION	CASS	9.99	8.49
THE ARK OF YESOD	CASS	7.95	6.75
TOLKIEN TRILOGY	CASS	12.99	11.04
TRAXEDON	CASS	9.99	8.49
TRAXEDON	DISC	14.95	12.70
VENGEANCE	CASS	9.99	8.49
WICKED	CASS	9.99	8.49
WICKED	DISC	14.99	12.74
WORLD TOUR GOLF	DISC	12.95	11.01

COMMODORE 64 ENTERTAINMENT

		R.R.P	OUR PRICE
WIZARD WARZ	CASS	9.95	8.46
WOLFMAN	CASS	9.99	8.49

AND MANY MORE TITLES PLEASE PHONE

EDUCaTIONAL SOFTWARE

		R.R.P	OUR PRICE
ALF IN THE COLOUR CAVES AGE 3-6	CASS	8.99	7.65
BETTER MATHS AGE 12-16	DISC	16.95	14.41
BETTER SPELLING AGE 6-16	DISC	16.95	14.41
BIOLOGY	DISC	16.95	14.41
BIOLOGY 1	CASS	10.95	9.31
BIOLOGY 2	DISC	16.95	14.41
BIOLOGY 2	CASS	10.95	9.31
CHEMISTRY	DISC	16.95	14.41
CHEMISTRY	CASS	10.95	9.31
CHEMIE+TRY 2	CASS	10.95	9.31
DANCE FANTASY AGE 4-8	CASS	8.99	7.65
FRENCH MISTRESS (ADVANCE)	CASS	8.95	7.61
FRENCH MISTRESS (BEGINNERS)	CASS	8.95	7.61
FUN SCHOOL 2 (Age 6-8)	CASS	9.95	8.46
FUN SCHOOL 2 (Age 6-8)	DISC	12.95	11.01
FUN SCHOOL 2 (Age 8+)	CASS	9.95	8.46
FUN SCHOOL 2 (Age 8+)	DISC	12.95	11.01
FUN SCHOOL 2 (Age Under 6)	CASS	9.95	8.46
FUN SCHOOL 2 (Age Under 6)	DISC	12.95	11.01
GEOGRAPHY/W.CLIMATE	DISC	16.95	14.41
GERMAN MASTER (Advance)	CASS	8.95	7.61
GERMAN MASTER (Beggiiners)	CASS	8.95	7.61
MAGIC MATHS (Age 4-8)	DISC	16.95	14.41
MATHOMIC (Age 12-17)	DISC	16.95	14.41
MATHS MANIA (Age 8-12)	CASS	17.25	14.66
OSPREY + Book	CASS	10.95	9.31
PHYSICAL GEOG (Age 12-17)	CASS	10.95	9.31
PHYSICAL GEOG (Age 12-17)	DISC	16.95	14.41
PHYSICS (Age 12-16)	CASS	10.95	9.31
PHYSICS (Age 12-16)	DISC	16.95	14.41
SPANISH TUTOR (Advance)	CASS	8.95	7.61
SPANISH TUTOR (Beggiiners)	CASS	8.95	7.61
WEATHER/CLIMATE	CASS	10.95	9.31
UP AND ADD'EM AGE 3-7	CASS	8.99	7.65

COMMODORE 64 BUSINESS & UTILITIES

		R.R.P	OUR PRICE
AMX MOUSE & STOP PRESS	DISC	69.95	62.95
AMX STOPPRESS SOFTWARE ONLY	DISC	39.95	33.96
AZIMUTH HEAD ALLIGNMENT	CASS	8.99	8.09
C64 DOCTOR		18.99	16.14
CASSETTE RECORDER By Cascade		24.95	21.20
CROS VERSION 1.3	DISC	29.95	25.46
MINI OFFICE 2	CASS	16.95	14.41
MINI OFFICE 2	DISC	19.95	16.96
SLIM-LINE KEYBOARD		19.95	17.96
POWER SUPPLY By Microbase		19.99	17.99
TASWORD 64	CASS	24.95	21.21
TASWORD 64	CASS	24.95	21.21
UTILITIES DISC V3	DISC	7.95	6.76

COMMODORE BUDGETS

	R.R.P	OUR PRICE
180	2.99	2.54
1942	2.99	2.54
720	2.99	2.54
ACE (BUDGET)	2.99	2.54
AIRWOLF 2	1.99	1.69
A.T.V.	2.99	2.54
BASEBALL (HIT SQUAD)	2.99	2.54
BATMAN	2.99	2.54
BMX SIMULATOR 2	2.99	2.54
BOULDERDASH 2	2.99	2.54
COBRA (HIT SQUAD)	2.99	2.54
COMIC PIRATE	2.99	2.54
COMMANDO	1.99	1.69
CRACK UP	2.99	2.54
CRAZY CARS	2.99	2.54
DANDY	2.99	2.54
DIE ALIEN SLIME	2.99	2.54
ENDURER RACER	2.99	2.54
FOOTBALL MANAGER	2.99	2.54
FREDDY HARDEST	2.99	2.54
FIRST MACHINE SIMULATOR	2.99	2.54
GRAND PRIX SIMULATOR	1.99	1.69
GOLD OR GLORY	1.99	1.69
HACKER 2	2.99	2.54
HYPER SPORTS	2.99	2.54
JOEY WILSON DARTS	1.99	1.69
KANE 2	2.99	2.54
KENDO WARRIOR	1.99	1.69
KIKSTART 2	2.99	2.54
KNIGHT GAMES	2.99	2.54
LAS VEGAS CASINO	2.99	2.54
LEADERBOARD	2.99	2.54
LEVIATHAN	2.99	2.54
LIGHT FORCE	2.99	2.54
MANIC MINER	2.99	2.54
MASTERS OF THE UNIVERSE	2.99	2.54
MASK	2.99	2.54
MEGA HITS	2.99	2.54
MEGA NOVA	2.99	2.54
METROCROSS	2.99	2.54
MISSION ELEVATOR	2.99	2.54
MISSION OMEGA	2.99	2.54
MOTORCROSS SIMULATOR	2.99	2.54
MOVING TARGET	2.99	2.54
NICK FALDO	2.99	2.54
NINJA COMMANDO	2.99	2.54
OCTUPLEX	2.99	2.54
ON CUE	2.99	2.54
ON THE BENCH	2.99	2.54
PITSTOP 2	2.99	2.54
POWERBOAT SIMULATOR	2.99	2.54
PRO MOUNTIAN BIKE SIMULATOR	2.99	2.54
RALLY	2.99	2.54
RAMPAGE	2.99	2.54
ROLLING THUNDER	4.99	4.24
ROLLING THUNDER DISC	2.99	2.54
SHORT CIRCUIT	1.99	1.69
SIGMA 7	1.99	1.69
SPACE ACADEMY	2.99	2.54
SPIN DIZZY	2.99	2.54
STREET FIGHTER	4.99	4.24
SUMMER GAMES DISC	2.99	2.54
THEATRE EUROPE	2.99	2.54
THUNDERCATS	2.99	2.54
TOP GUN	2.99	2.54
TREASURE ISLAND DIZZY	2.99	2.54
TURBO BIKE	2.99	2.54
WAR MACHINE	2.99	2.54
WIZBALL	2.99	2.54
WOLF PACK	2.99	2.54
WORLD GAMES	2.99	2.54
WORLD TOUR GOLF	2.99	2.54
YIE AR KUNG FU	2.99	2.54
ZAMZARA	2.99	2.54
ZYNEX	2.99	2.54

PLEASE STATE MAKE OF COMPUTER

PLEASE STATE MAKE OF COMPUTER

PLEASE STATE MAKE OF COMPUTER

PLEASE STATE MAKE OF COMPUTER

NEWS FLASH THE CD GAMES PACK By Codemaster 30 GAMES + SOFTWARE + CABLE ONLY £19.95 For Commodore 64 - Amstrad CPC - Spectrum

Techno Info

Our resident boffin sifts through some of your questions and tries to come up with the answers

Dear CDU,
Recently I purchased a STAR LC-10 Colour Printer as I was told that several of my disk-based art packages supported colour printers. However, after weeks of trying I am still unable to find suitable configurations for the disks. I have copies of OCP Art Studio, Print Master, Print Shop and News Room. I returned the printer to the dealer although after testing he told me that there was nothing wrong as the test program in the manual printed in colour. I would appreciate any help or information you can give me whether it be on how to configure the disks I have or suggest any software colour compatible with the STAR LC-10C.
Mr C. Best, Nottingham.

Dear Mr. Best,
From the information you have given about the test carried out when you returned the printer to the dealer, it is obviously working all right. I believe that none of the art packages you mentioned that are for the commodore 64 support any colour printer. Most art packages that print in colour are directed towards the more powerful computers such as the Amiga and there are in fact very few software packages that have options or possible configurations for printing in colour. One that I do know of is 'Award Maker Plus 64', an American program imported by F.S.S.L. of 18 High Street, Pershore, Worcestershire. Their telephone number is (0836) 553153 and the program retails at about twenty-five pounds. I have an assurance from F.S.S.L. that the program is colour compatible with the STAR LC-10. I hope this information will be of use to you and I would like to hear how this issue progresses as I am sure that there are other people in the same situation.

Dear CDU,
I have just one question - please could you tell me how to obtain the 'up and down' rolling effect on the Commodore Disk User logo on the menu screen as I have been wanting to do this for a number of months.
Michael Pitches, Plymouth.

Dear Michael,
The colour rolling effect is quite difficult to achieve unless, that is, you have a knowledge of how to manipulate each individual line of the screen by the use of raster interrupts. Andy Partridge produced an excellent set of routines in the November 1989 issue of Commodore Disk User and I would suggest that you look these out. Instead of changing the border and background colours, try changing multicolour registers one and two (53282 and 53283). You then use multicolour characters and any with colour scrolling

downwards uses register one for its data and any upwards uses register two. Of course, you will have to add a routine to move the colour table about in memory (or at least which byte in the table is used as the first colour each time). You could alternatively, if you have a machine language monitor, hack into the 'CDU Menu Kit' provided with the December issue. The relevant code is at $3625 onwards (lookout for the 53282 and 53283). I hope you will find this information of some assistance.

Dear CDU,
With reference to the Hi-Res Demo Kit by Neil Higgins (Jul/Aug 1989), could you please tell me how to load a BASIC program automatically once the space bar has been pressed. I do not know any machine code and so would be grateful of any practical information you could offer. Secondly, do you know of any machine code books and general users books that would be worth my while purchasing?
Scott Mathieson, Hamilton, Scotland.

Dear Scott,
On the disk you will find a BASIC program (filed as "PROB1") that will read some machine code data and then save it. Prior to the saving you must enter the name of the actual demo created using the kit, followed by the name of the BASIC program that you want to auto-run. Then after that, each time you wish to load the chosen demo you will instead type LOAD"DEMO LOADER",8 and then RUN. Your chosen demo will then be loaded and executed. Now, when the spacebar is pressed to end the demo, the program (with the filename that you will have specified when the BASIC machine code loader was used) will be loaded and RUN automatically. You should note that the demo loader, actual demo and secondary BASIC program should all be on the same disk. I hope you will find this of help. Now on to the some books. The one that started me off on machine code was 'An Introduction To Machine Code on the Commodore 64" although there are quite a number of books on the market. The latter provides you with all the details to get started but practice and investigation are the key. Buy yourself a decent assembler (or use the 6510+ published in CDU a little while back). These will help you to write, at first quite simple, but also more complex machine code programs as you become more confident. I find the "Commodore 64 Programmer's Reference Guide" published by Commodore Business Machines and "Using the 64" by Peter Gerrard and published by Duckworth Home Computing are also very good general books, providing information for both the beginner and the expert of the latter two also contain information on machine code programming and the Programmers Reference Guide is available from most shops that deal mainly with home computers.

Dear CDU,
I have recently bought the ADVANCED ART STUDIO and want to use the pictures with my programs. A loader is provided with the package although it simply states that by listing the program you will be able to modify it so that it works with your own programs. The standard loader uses screen at 1024 and loads the picture data to 8192. If I want to move this to bank 2 then it all gets tangled up with the ROM at 40960 ($A000). I have raised the start of memory and I have moved the screen from 1024. I have tried all sorts of experiments but with little result. I could do with some help on this one because the quality of the graphics package is worthwhile but there must be a better way to make use of the RAM. Also, on my Vic 20 I had a routine that could make a screen slide out of the bottom of the screen and change to the next screen and slide back in. This could be done in BASIC very smoothly and I wondered if you knew how to do this on the '64. I should mention that I know nothing of SYS commands and machine code.
Doug Sneddon, Wilsthire.

Dear Doug,
I understand how you feel, but most of these picture loaders are designed so that beginners can easily relate the method of displaying the picture to those outlined in many manuals – i.e. with the bitmap starting at 8192. This severely limits the length of the BASIC program area to about 6K unless you start playing about with pointers. It is possible to display the bitmap with the data beginning at $E000. With this method, utilising Bank 3, colour memory remains at $D800 and the screen memory is moved to $C800. This does not restrict the BASIC memory area at all. You change the computer to this format with the following POKEs: 53272,41 – 53265,189 – 53270,216 – 56576,128. However, because the region for the bitmap is under the ROM used for input/output device calls it is advisable (because you will need to save the picture data) to store the information at $A000 to $C800 and then convert each block of information to the correct addresses upon loading. This would take a very long time in BASIC and therefore, although you mention you do not understand machine code, I have written a program in machine code that will do all this for you. On the disk you will find the program "PROB2". Load it like a normal BASIC program and then RUN it. You will be requested for the name of your original picture (you should omit the MPIC suffix). This will then be loaded, converted to the new format at $A000 and resaved under the name "PICTURE" which you may of course change using the RENAME command. You will then have the option of saving the display code. This is a small chunk of code that resides in the tape buffer and is used to transfer all the information from $A000 to $E000 and to then display the picture. It may seem a little complicated but if you follow these instructions you shouldn't go wrong: To load a picture back, first load "DISPLAY CODE" with the ,8,1 suffix. This need only be done once. Then load the 'new' picture as usual. To display it simply enter SYS828. This could easily be incorporated in a BASIC program. I appreciate that you don't understand SYS calls but I'm afraid this is a situation where you will simply have to accept that they work. Your second query is slightly more difficult and unfortunately I am unaware of such a BASIC routine on the 64. However, in a future issue of CDU I hope to be presenting a smooth screen slide of this sort but written in machine code. I hope that I have been of assistance.

That just about sums up this months batch of problems. Don't forget, if you have a problem, and no-else can help, perhaps you could call the 'Techno Info' team.

COMMODORE Disk User

FOR C64 AND C128 USERS

Lineage: 53p per word. (+ VAT)
Semi display: £11.50 plus VAT per single column centimetre minimum 2cm. Ring for information on series bookings/discounts.

All advertisements in this section must be prepaid.
Advertisements are accepted subject to the terms and conditions printed on the advertisement rate card (available on request).

Send your requirements to:
CLASSIFIED DEPARTMENT
ASP LTD. ARGUS HOUSE,
BOUNDARY WAY, HEMEL HEMPSTEAD HP2 7ST.

0442 66551

REPAIRS

COMMODORE SPARES & REPAIRS

REPAIRS

C64, C+4 ..... £25 inc.
C128, 1541 ..... £35 inc.

SPARES

906114	£9.30	901227	£11.95
6510	£9.95	8501	£11.45
6526	£10.95	C64 Power Pack	£22.00
6581	£12.95	C64 Cassette Unit	£22.00
6569	£19.95	C128 Power Pack	£69.95

Inc Vat & PP

OASIS COMPUTER SERVICES,
Dept 4C, 14 Ridgeway Road, Salisbury, Wiltshire, SP1 3BU.
Tel (0722) 335061

SOFTWARE

PUBLIC DOMAIN SOFTWARE for the C64/128.
We have 150 disks full of all types of Programs
All disks are £2.95 each
NOW FOR THE +4
from £3.75 per disk.
Send SAE or phone for free catalogue (state model)
Kingsway Computer Services
72 Glencoe Road, Sheffield
S2 2SR Tel: (0742) 750623

TO ADVERTISE IN CDU CALL ANTHONY FLANAGAN ON 0442 66551

ONLY POOLS AND HORSES

FOOTBALL BOXFORM £14.95 (+£2 for 3" disc)

Not a gimicky plaything but a genuine statistical analysis of football form the sole objective of which is to improve the chances of winning on the pools or fixed odds. Written by a pools expert it has forecast over 50% more draws than would be expected by chance. Homes aways and draws are shown in order of merit and true odds given for every match. Merit tables show at a glance the teams currently in form and those having a lean spell.

BOXFORM & HANDICAP WINNER £14.95 (+£2 for 3" disc)

Two programs for the price of one. **BOXFORM** weighs up a race by considering the many facets of form. Easily amended to suit the user's ideas. **HANDICAP WINNER** shows the value for money bets. Over 1000 winners in 1989 handicaps at 28/1, 25/1, 20/1, 16/1, 15/1.....etc.
SPECIAL OFFER: £24.95 (+£4 for 3" discs) for all the above programs plus a **FREE BETS CALCULATOR** that's so good it's used by bookies.

RACING TOOL KIT (£49.95 tape or disc)

A package of 7 programs for the racing enthusiast. BOXFORM and HANDICAP WINNER plus programs to produce ratings and time figures, update private handicaps or form your own, keep records, work out almost any bet, etc.
Why pay inflated prices? BOXoft CLEVER and get the BEST
Available on tape or disc for SPECTRUM, CMD C64/128, AMS CPC's, BBC B. from:

BOXOFT (CDU), 65 Allans Meadow, Neston, South Wirral. L64 9SQ
Credit card holders may phone 051-336-2668 (24 hrs)

REPAIRS

SAME DAY SERVICES (CALLERS) C64 ONLY

HINDLEY ELECTRONICS

FAST

24 hr TURNAROUND (POSTAL) C64 ONLY

EXPERT COMPUTER REPAIRS
COMMODORE - SINCLAIR - ACORN - AMSTRAD

STANDARD CHARGES
(INC VAT AND FULLY INSURED RETURN POSTAGE
FOR NEXT DAY DATAPOST DELIVERY ADD £14)

COMMODORE 64	£30	COMMODORE 128	£35	COMMODORE C16	£22
COMMODORE 64C	£32	COMMODORE +4	£28	COMMODORE VIC 20	£18

AMIGA 500 £40 (excluding disk drive)
Please send power unit with Computer
IF YOUR COMPUTER IS NOT LISTED - RING FOR DETAILS

1540 - 41 disk drive	£35	1571 disk drive	£40
1570 disk drive	£35	AMIGA 500 disk drive replacement	POA

PRINTERS - PLEASE RING FOR DETAILS

3 MONTHS "NO QUIBBLE" WARRANTY ON ALL REPAIRS TRADE REPAIRS UNDERTAKEN
LOW PRICES - SPARES - SPARES - SPARES - LOW PRICES
COMPUTER ICS AND GENERAL COMPONENTS *WHY PAY MORE*

HINDLEY ELECTRONICS
DEPT 102, 97 MARKET STREET, HINDLEY WIGAN WN2 3AA Tel: (0942) 522743

Computer repairs: Callers requiring same day service please ring before calling and add 10% to standard charge - Standard charges do not include repairs to external items E.G. power units tape recorders joysticks etc. A surcharge may be levied for machines that have sustained severe damage E.G. fire flood tamper

ADVERTISERS

CLASSIFIED COUPON

ALL CLASSIFIED ADVERTISEMENTS MUST BE PRE-PAID. THERE ARE NO REIMBURSEMENTS FOR CANCELLATIONS.
CLASSIFIED DEPT. ARGUS HOUSE, BOUNDARY WAY, HEMEL HEMPSTEAD HP2 77ST.
Lineage: 53p per word. (+ VAT) Semi display: £11.50 plus VAT per single column centimetre minimum size 2cm. Series discounts available.

I enclose my Cheque/Postal Order for £ .....for .....
insertions, made payable to Argus Specialist Publications.
(Delete as necessary)

PLEASE DEBIT MY ACCESS/BARCLAYCARD NO

EXP. DATE .....

£.....**FOR**.....**INSERTIONS**

Name.....

.....**Post Code**.....

Daytime Tel No......

Signature.....**Date**.....

☐ FOR SALE ☐ SOFTWARE ☐ SPECIAL OFFERS ☐ REPAIRS

CALL 0442 66551 TO ADVERTISE

M.C.E. SERVICES
33, Albert Street, Mansfield, Notts. NG18 1EA Tel: 0623512
FIXED PRICE COMMODORE REPAIRS

C64	£24.95
C16	£19.95
PLUS4	£24.95
AMIGA A.500	£29.95
1541	£29.95
1571	£31.95
C2N etc	£12.95

All prices inc. VAT, Labour, Parts etc.
NO MORE TO PAY.
Please ring for details.

COMMODORE REPAIRS, SPARES & LEADS
C64, +4, 16 - **£24.00** C128 - **£32.00**
AMIGA 500 -£34.00
Three month warranty. For price list send 1st class stamp (State model)
ACE REPAIRS, Outways FM, Pelynt, Looe, Cornwall, PL13 2NW. Tel. (0503) 20282

A1 COMPUTER SERVICES
Unit 9 Paddock Mount Offices, Dawley, Telford TF4 3PR
Tel: 0952 502737.
Computer repairs, supplies, leases & credit available.
Credit Brokers.

THE STRATEGIST

Gordon Hamlett continues his column on strategy for the home computer

When I was at school, there was a teacher who, it was rumoured, never actually read our history projects but weighed them instead. By a similar token, *Omega*, the latest game from **Origin** must be the best title released for a long time.

For anyone who likes loading in a game and playing it straight away is soon going to come to grief here. The instruction manual is over 200 pages long! That is aside from a couple of trifling training and reference manuals.

Tank design is the name of the game. You have just started a new job at the Organisation of Strategic Intelligence, the world's leading manufacturer of state of the art cybertanks. Not only must you put together the hardware, keeping within your budget limitations, but you also have to write the software that controls the tank on the field of battle. You will then have to test your design on a simulated field of battle and only if your tank is successful in at least 70% of its combats will you be granted a higher security clearing and additional funds.

Selecting the hardware is the least of your problems. Body, weaponry, drive system, fuel tank and scanner are the five compulsory items and you have several choices within each category. In addition, there are several optional extras such as early warning systems and repair kits although you are not likely to be able to afford them for some time.

Basically, the more you pay, the better you get but it would be wrong to assume that you should strive for the best of everything. You might prefer a light fast tank to a slow heavy one or your tactics might be better suited to a weapon with a fast reload time at the expense of greater damage.

The main part of the game is concerned with providing your tank with the 'artificial intelligence' necessary to overcome its adversaries on the battlefield. The language used is a fairly simple structured one making good use of named procedures or subroutines.

Every effort has been made to make life easy for the user. To this end, there are a series of key words at the bottom of the edit screen that can be clicked on in order to insert them directly into your program. In addition, several of them will open up a new menu or prompt you for the likely continuation. All of this is designed to help you get the syntax correct even if your logic proves to be faulty.

There are several pre-written capsules that you can include in your program that removes the need to work out certain routines for yourself. These may not prove to be exactly what you want but there should be enough material here to keep you quiet for a while.

The basic requirements for your tank are to move, find the enemy and then destroy it so a very simple program might look like the following where *SEEK* and *DESTROY* are named routines;

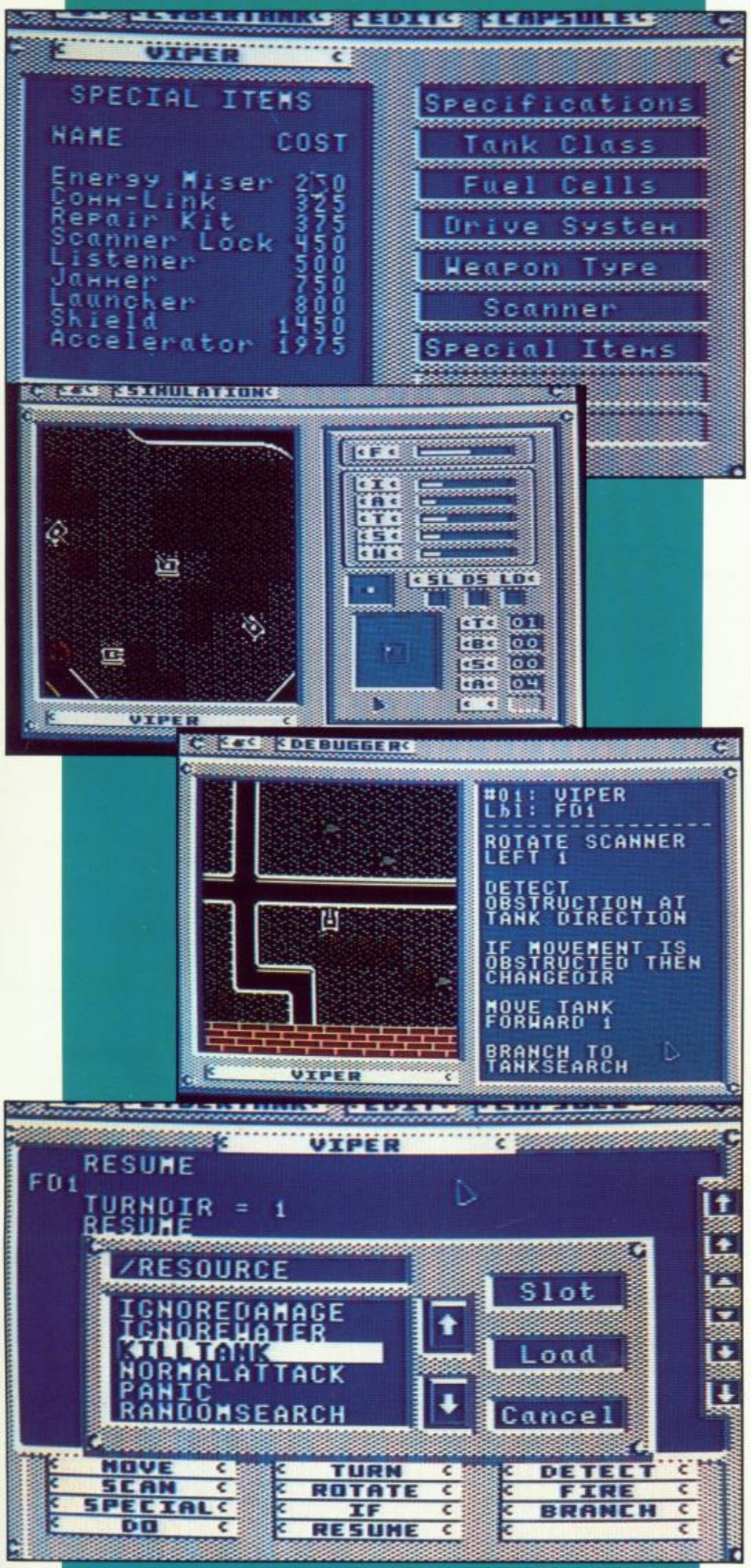
```
START
DO SEEK
DO DESTROY
BRANCH TO START

INCLUDE SEEK
INCLUDE DESTROY
```

As you progress, so you will want to include conditional statements in your design and make use of the many system variables. A typical statement might be; *If enemy tank is beyond weapon range then branch to smart-move or if scanner is not aligned with tank then rotate scanner left 3.*

What the capsules will not do for you is decide on your tactics. It is up to you to tell your tank when to run away, when to hide and how best to conduct an attack – stand and slug it out or hit and run. You will also have to manoeuvre round indestructable objects and optimise your search routines. Remember, it is just as important to pick up the tank sneaking up behind you as it is to destroy those in front of you. You will also have to write your own routines for most of the optional extras – when to raise your shields, when to launch the remote satellite and so on.

There are full editing facilities and you will also have the opportunity to step through and put a trace on your program as you watch your tank in battle.



Once you think you have a reasonable design, it is time to test it. First though, your design must be authorised which is a basic check to ensure that all the parts are present and the syntax is correct.

There are three battlefields included in the package or you can design your own. All you have to do is specify the numbers of enemy tanks you want plus the types. At the start, it is everyone for themselves, but at advanced levels, you can opt for team play providing that you set up the necessary communication links in your program.

The ensuing battle shows a top down view of the area immediately surrounding your tank. On the right of the screen are various indicators showing the amounts of damage sustained to different parts of your tank although these indicators can be replaced by a copy of your program if you are in debug mode. In addition, you can switch to a satellite picture showing the entire field of combat.

When you feel happy with your design, you can put it to the test by requesting a clearance evaluation, the program will then assess your tank's performance in ten successive battles. Win seven and you are promoted. Lose and it is back to the drawing board.

Obviously, a game of this complexity stands or falls according to the quality of its documentation and in this aspect, Omega succeeds admirably. There is a lot of unnecessary jargon but that aside, the manual takes you on a step by step tutorial through the program as you get to grips with it. The on screen commands are all easy to use being menu driven. The only problem I had was when I was not prompted to swap disks on one occasion. Graphically, *Omega* is no great shakes but that has little bearing on the game itself.

This is the sort of game that you will either love or hate. For those that fancy themselves as great tacticians and masters of logic, I can foresee many a long night hunched over the keyboard as this will probably be the only game that they need to buy in the coming year.

Title: Omega
Supplier: Origin via Mindscape
Price: £19.99

...it's dynamite!

POWER CARTRIDGE

FOR YOUR COMMODORE 64/128

..."unbelievable value for money" ZZAPP! Dec 89

- POWER TOOLKIT
- POWER MONITOR
- TAPE & DISK TURBO
- PRINTERTOOL
- POWER RESET
- TOTAL BACKUP

..."Money well spent" YC/CDU Jan 90

42 Pg Manual - "Damned Good Handbook" CCI Jan 90

SO MUCH FOR SO LITTLE

AVAILABLE FROM ALL GOOD COMPUTER RETAILERS

TRIED AND TESTED - OVER 100,000 SOLD IN EUROPE

...."highly recommended for C64 users" CCI Jan 90

YOU WILL WONDER HOW YOU EVER MANAGED WITHOUT IT

POWER TOOLKIT

A powerful BASIC-Toolkit (Additional helpful commands) that considerably simplifies programming and debugging.

AUTO	HARDCAT	RENUMBER
AUDIO	HARDCOPY	REPEAT
COLOR	HEX$	SAFE
DEEK	INFO	TRACE
DELETE	KEY	UNNEW
DOKE	PAUSE	QUIT
DUMP	PLIST	MONITOR
FIND	ILOAD	BLOAD

RENUMBER - *Also modifies all the GOTO's GOSUB's etc. Allows part of a program to be renumbered or displaced.*

PSET - *Set up of printer type.*

HARDCAT - *Prints out Directory.*

The toolkit commands can be used in your programs.

DISK TOOL

Using POWER CARTRIDGE you can load up to 6 times faster from disk.
The Disk commands can be used in your own programs.

DLOAD	DVERIFY	DIR
DSAVE	MERGE	DEVICE
DISK		

MERGE - *Two BASIC programs can be merged into one.*

DISK - *With DISK you can send commands directly to your disk.*

TAPE TOOL

Using POWER CARTRIDGE you can work up to 10 times faster with your data recorder. The Tape commands can be used in your own programs.

LOAD	SAVE	VERIFY
MERGE	AUDIO	

POWERMON

A powerful machine language monitor that is readily available and leaves all of your Commodore memory available for programming.
Also works in BASIC-ROM, KERNAL and I/O areas.

A ASSEMBLE	I INTERPRET	S SAVE
C COMPARE	J JUMP	T TRANSFER
D DIS-ASSEMBLE	L LOAD	V VERIFY
	M MEMORY	W WALK
F FILL	P PRINT	X EXIT
G GO	R REGISTER	$ DIRECTORY DOS Commands
H HUNT		

PRINTERTOOL

The POWER CARTRIDGE contains a very effective Printer-Interface, that self detects if a printer is connected to the Serial Bus or User-Port.
It will print all Commodore characters on Epson and compatible printers.
The printer-interface has a variety of set-up possibilities. It can produce HARDCOPY of screens not only on Serial printers (MPS801, 802, 803 etc) but also on Centronic printers (EPSON, STAR, CITIZEN, PANASONIC, etc).
The HARDCOPY function automatically distingishes between HIRES and LORES. Multi-colour graphics are converted into shades of grey. The PSET functions allow you to decide on Large/Small and Normal/Inverse printing.
The printer PSET functions are:

PSET 0 - *Self detection Serial/Centronics.*
PSET 1 - *EPSON mode only.*
PSET 2 - *SMITH-CORONA mode only.*
PSET 3 - *Turns the printing 90 degrees!!*
PSET 4 - *HARDCOPY setting for MPS802/1526.*

PSET 8 - *Bit-image mode.*
PSET C - *Setting Lower/Upper case and sending Control Codes.*
PSET T - *All characters are printed in an unmodified state.*
PSET U - *Runs a Serial printer and leaves the User-port available.*
PSET Sx - *Sets the Secondary address for HARDCOPY with Serial Bus.*
PSET L1 - *Adds a line-feed, CHR$ (10), after every line.*
PSET L0 - *Switches PSET L1 off.*

POWER RESET

On the back of the POWER CARTRIDGE there is a Reset Button. Pressing this button makes a SPECIAL MENU appear on the screen.
This function will work with any programme.

CONTINUE - *Allows you to return to your program.*
BASIC - *Return to BASIC.*
RESET - *Normal RESET.*
TOTAL BACKUP - *Saves the contents of the memory onto a Disk. The program can be reloaded later with BLOAD followed by CONTINUE.*
DISK RESET ALL - *RESET of any program.*
TOTAL BACKUP TAPE - *As BACKUP DISK but to TAPE.*
HARDCOPY - *At any moment, prints out a Hardcopy of the screen. Using CONTINUE afterwards you can return to the program.*
MONITOR - *Takes you into the Machine language Monitor.*

BDL

Bitcon Devices Ltd

88 BEWICK ROAD
GATESHEAD
TYNE AND WEAR
NE8 1RS
ENGLAND

Tel: 091 490 1975 and 490 1919 Fax 091 490 1918
To order: Access/Visa welcome - Cheques or P/O payable to BDL
Price: £16.99 incl. VAT.
UK orders add £1.20 post/pack total - £18.19 incl. VAT.
Europe orders add £2.50. Overseas add £3.50
Scandinavian Mail Order and Trade enquiries to: Bhiab Elektronik, Box 216, Norrtalje 76123, SWEDEN. Tel: + 46 176 18425 Fax: 176 18401
TRADE AND EXPORT ENQUIRIES WELCOME